Petr Sojka
Aleš Horák
Ivan Kopeček
Karel Pala (Eds.)

LNAI 6231

Text, Speech and Dialogue

13th International Conference, TSD 2010
Brno, Czech Republic, September 2010
Proceedings

Lecture Notes in Artificial Intelligence 6231

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Petr Sojka Aleš Horák
Ivan Kopeček Karel Pala (Eds.)

Text, Speech and Dialogue

13th International Conference, TSD 2010
Brno, Czech Republic, September 6-10, 2010
Proceedings

Springer

Volume Editors

Petr Sojka
Masaryk University
Faculty of Informatics
Department of Computer Graphics and Design
Botanická 68a, 602 00 Brno, Czech Republic
E-mail: sojka@fi.muni.cz

Aleš Horák
Ivan Kopeček
Karel Pala
Masaryk University
Faculty of Informatics
Department of Information Technologies
Botanická 68a, 602 00 Brno, Czech Republic
E-mail: {hales,kopecek,pala}@fi.muni.cz

Library of Congress Control Number: 2010933819

CR Subject Classification (1998): I.2, H.3, H.4, H.5, J.1, H.2

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-15759-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15759-2 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The annual Text, Speech and Dialogue Conference (TSD), which originated in 1998, is now starting its second decade. So far more than 1,000 authors from 45 countries have contributed to the proceedings. TSD constitutes a recognized platform for the presentation and discussion of state-of-the-art technology and recent achievements in the field of natural language processing. It has become an interdisciplinary forum, interweaving the themes of speech technology and language processing. The conference attracts researchers not only from Central and Eastern Europe but also from other parts of the world. Indeed, one of its goals has always been to bring together NLP researchers with different interests from different parts of the world and to promote their mutual cooperation. One of the ambitions of the conference is, as its title says, not only to deal with dialogue systems as such, but also to contribute to improving dialogue between researchers in the two areas of NLP, i.e., between text and speech people. In our view, the TSD Conference was successful in this respect again in 2010.

This volume contains the proceedings of the 13th TSD Conference, held in Brno, Czech Republic in September 2010. In the review process, 71 papers were accepted out of 144 submitted, an acceptance rate of 49.3%. We would like to thank all the authors for the efforts they put into their submissions and the members of Program Committee and reviewers who did a wonderful job helping us to select the most appropriate papers. We are also grateful to the invited speakers for their contributions. Their talks provide insight into important current issues, applications and techniques related to the conference topics.

Special thanks are due to the members of Local Organizing Committee for their tireless effort in organizing the conference.

The TEXpertise of Petr Sojka resulted in the production of the volume that you are holding in your hands.

We hope that both participants and readers will benefit from the results of this event and disseminate the ideas of the TSD Conference all over the world.

July 2010

Aleš Horák
Ivan Kopeček
Karel Pala
Petr Sojka

Organization

TSD 2010 was organized by the Faculty of Informatics, Masaryk University, in cooperation with the Faculty of Applied Sciences, University of West Bohemia in Plzeň. The conference website is located at www.tsdconferences.org/tsd2010/

Program Committee

Jelinek, Frederick (USA), *General Chair*
Hermansky, Hynek (USA), *Executive Chair*
Agirre, Eneko (Spain)
Baudoin, Geneviève (France)
Černocký, Jan (Czech Republic)
Ferencz, Attila (Romania)
Gelbukh, Alexander (Mexico)
Guthrie, Louise, (UK)
Hajič, Jan (Czech Republic)
Hajičová, Eva (Czech Republic)
Hanks, Patrick (Czech Republic)
Hitzenberger, Ludwig (Germany)
Hlaváčová, Jaroslava (Czech Republic)
Horák, Aleš (Czech Republic)
Hovy, Eduard (USA)
Kopeček, Ivan (Czech Republic)
Krauwer, Steven (The Netherlands)
Kunzmann Siegfried (Germany)
Loukachevitch, Natalija (Russia)
Matoušek, Václav (Czech Republic)
Ney, Hermann (Germany)
Nöth, Elmar (Germany)
Oliva, Karel (Czech Republic)
Pala, Karel (Czech Republic)
Pavesić, Nikola (Slovenia)
Petkevič, Vladimír (Czech Republic)
Pianesi, Fabio (Italy)
Przepiorkowski, Adam (Poland)
Psutka, Josef (Czech Republic)
Pustejovsky, James (USA)
Rothkrantz, Leon (The Netherlands)
Rusko Milan (Slovakia)
Schukat-Talamazzini, E. Günter (Germany)
Skrelin, Pavel (Russia)
Smrž Pavel (Czech Republic)
Sojka Petr (Czech Republic)
Tadić, Marko (Croatia)
Varadi, Tamas (Hungary)
Vetulani, Zygmunt (Poland)
Vintsiuk, Taras (Ukraine)
Wilks, Yorick (UK)
Zakharov, Victor (Russia)

Referees

Štefan Beňuš, Wauter Bosma, Miloš Cerňak, Radoslav Forgáč, Nestor Garay-Vitoria, Maria Khokhlova, Daniil Kocharov, Olga Kolesnikova, Richard Kováč, Oier Lopez de Lacalle, Yulia Ledeneva, Iker Luengo-Gil, Tetyana Lyudovyk, Olga Mitrofanova, Ilya Oparin, Piotr Pęzik, Valeriy Pylypenko, Anna Rumshisky, Róbert Sabo, Mykola Sazhok, Pavel Schlesinger, Petr Schwarz, Aitor Soroa, Barbora Vidová Hladká, Nina Volskaya

Organizing Committee

Dana Hlaváčková *(Administrative Contact)*, Aleš Horák *(Co-chair)*, Dana Komárková *(Secretary)*, Ivan Kopeček, Karel Pala *(Co-chair)*, Adam Rambousek *(Web System)*, Pavel Rychlý, Petr Sojka *(Proceedings)*

Sponsors and Support

The TSD Conference is regularly supported by the International Speech Communication Association (ISCA). We would like to express our thanks to Lexical Computing Ltd., for their kind sponsorship of TSD 2010.

Table of Contents

I Invited Papers

II Text

III Speech

IV Dialogue

Part I

Invited Papers

Parsing and Real-World Applications

John Carroll

School of Informatics, University of Sussex
Falmer, Brighton BN1 9QJ, UK
J.A.Carroll@sussex.ac.uk
http://www.informatics.sussex.ac.uk/research/groups/nlp/carroll/

Abstract. Much recent research in natural language parsing takes as input carefully crafted, edited text, often from newspapers. However, many real-world applications involve processing text which is not written carefully by a native speaker, is produced for an eventual audience of only one, and is in essence ephemeral. In this talk I will present a number of research and commercial applications of this type which I and collaborators are developing, in which we process text as diverse as mobile phone text messages, non-native language learner essays, and primary care medical notes. I will discuss the problems these types of text pose, and outline how we integrate information from parsing into applications.

Keywords: Natural language, parsing, ephemeral text, applications.

1 Introduction

The creation of large treebanks of newspaper text and other edited text genres has been very successful in driving forward natural language parsing technology. However, parsers trained on these treebanks often fail to produce useful output when confronted with telegraphic or other types of non-standard text. Moreover, even when parsers can be applied successfully to such text, it is often not obvious within an application context how to leverage the grammatical information produced.

In this talk I will present some research and commercial applications that I and collaborators are developing, in which the type of language to be processed is 'noisy' and very different to edited, newspaper text. We are carrying out this work through the company iLexIR [1] and through funded research projects with commercial partners.

In these applications we are building on two basic language processing tools, RASP and TAP:

- RASP [2] is a robust, domain-independent analysis system for English text, comprising modules for finding sentence boundaries, word tokenisation, morphological analysis, assigning part-of-speech labels, and analysing the grammatical relations between words and larger units within sentences. The system is accurate and efficient, and is embedded in an XML pipeline.
- The Timed Aggregate Perceptron (TAP) classifier [3] is a highly scalable linear classifier which has been shown to outperform SVMs and Bayesian logistic regression on topic and other text classification tasks. The TAP classifier achieves better classification accuracy than either popular alternative, and moreover can be trained in near linear time.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 3–5, 2010.

2 Applications

2.1 Automated Assessment of Examination Scripts

Teaching and assessment of non-native English language learners is a highly competitive and valuable area of business. iLexIR has built a system for Cambridge Assessment (a major international exams group) which automatically marks certain types of English language learner essays. We formulate marking as a classification problem, training on documents (essays) and manual classifications (examiners' marks), extracting features from the documents (analysed with RASP), and using machine learning to determine feature weights which correlate with the manual classification. In terms of marking performance, the current version of the system is practically indistinguishable from an experienced human examiner.

2.2 SMS Question Answering

A number of commercial SMS any-question services have been set up in recent years, accepting text message questions from the general public and charging a fee for sending back an answer. A significant proportion of questions are directory enquiries of some sort, for example:

> Q: *Vets, market street in rugeley*
>
> A: *Donnachie & Townley, The Veterinary Centre, Market St, Rugeley, Staffordshire, WS15 2JH, T: 01889582023.*

We have used RASP and TAP to build a system for one such service which classifies incoming SMS questions into enquiry types, extracts the relevant information from the questions, and builds structured queries which can be run against an appropriate directory to produce results which are used to semi-automatically construct answers. The system has delivered significant cost savings to the service provider.

2.3 Mining Text in Electronic Patient Records

In the UK, all primary care patient consultations are computerised, and large databases of patient records have been collected and are being used for pharmacovigilance, disease surveillance, and health services research. In a research project funded by the Wellcome Trust, we are working with epidemiologists to extract information about signs and disease symptoms from the text that doctors type. Although some of this text is so telegraphic as to be almost unparsable in principle, we have had success in using parser output to automatically create distributional thesauruses [4] in order to identify expressions with related meanings. In preliminary work, we have been able to find many cases of a certain disease that were not originally coded as such.

3 Conclusions

In the systems we have built the way in which parsing is integrated varies widely, but the results demonstrate that parsing can be useful in a diverse set of language-processing applications involving 'noisy' input.

Acknowledgments. Many thanks to my collaborators Ted Briscoe, Øistein Andersen, Paula Buttery, Rob Koeling, Ben Medlock, and Rebecca Watson, who have led or contributed to much of the work I am presenting.

References

1. iLexIR, http://www.ilexir.com
2. Briscoe, E., Carroll, J., Watson, R.: The Second Release of the RASP System. In: COLING/ACL 2006 Interactive Presentation Sessions, pp. 77–80. Association for Computational Linguistics(2006)
3. Medlock, B.: Investigating Classification for Natural Language Processing Tasks. VDM Verlag (2008)
4. Weeds, J., Weir, D.: Co-occurrence Retrieval: a General Framework for Lexical Distributional Similarity. Computational Linguistics 31(4), 439–476 (2005)

Knowledge for Everyman
(Extended Abstract)

Christiane Fellbaum

Department of Computer Science, Princeton University
35 Olden Street, Princeton, NJ 08540, USA
fellbaum@princeton.edu

Abstract. Increasing globalization creates situations with wide-ranging effects on large communities, often requiring global responses and innovative solutions. Timely examples are climate and environmental changes related to rapid growth and economic development. Natural and man-made unforeseen catastrophes like oil spills, landslides and floods require immediate action that might crucially rely on information and expertise available only from sources far removed from the crisis site. Knowledge sharing and transfer are also essential for sustainable long-term growth and development. In both kinds of cases, it is important that information and experience be made available and widely shared, communicated and encoded for future re-use. The global scope of many problems and their solutions requires furthermore that information and communication be accessible to communities crossing languages and cultures. Finally, an appropriate system for recording, maintaining and sharing information must be accessible to both experts and laymen.

The goal of the European Union-funded KYOTO project (Knowledge-Yielding Ontologies for Transition-Based Organization, http://www.kyoto-project.eu) is to develop an information and knowledge sharing system that relates documents in several languages to lexical resources and a common central ontology and allows for deep semantic analysis. KYOTO facilitates the crosslinguistic and crosscultural construction and maintenance of a sophisticated knowledge system among the members of domain-specific communities. Representation, storage and retrieval of a shared terminology takes place via a Wiki platform. Relevant terms are anchored in a language-independent, customizable formal ontology that connects the lexicons of seven languages (Basque, Chinese, Dutch, English, Italian, Japanese, and Spanish) and that guarantees a uniform interpretation of terms across languages. The semantic representations in the ontology are accessible to a computer and allow deep textual analysis and reasoning operations.

KYOTO's target domains are the environment and biodiversity, with appropriate experts acting as "users". Once developed, the system will be available for extension to any domain.

Keywords: Crosslingual wordnets, knowledge representation system, ontology, text mining.

1 KYOTO's Architecture

Among KYOTO's principal components are linguistic text miners, a Wiki environment for supporting and maintaining the system, and a portal for the environmental and

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 6–9, 2010.

biodiversity domains that allows for deep semantic searches. Concept extraction and data mining are applied through a chain of semantic processors (“Kybots”) that share a common knowledge base and re-use the knowledge for different languages and particular domains. Information access is provided through a crosslinguistic user-friendly interface that allows for high-precision search and information dialogues for a variety of data from different sources. The system is maintained and updated by specialists in the field using the open Wiki platform for ontology maintenance and wordnet extension.

2 Domain-Specific Wordnet and Ontology Construction in the KYOTO System

Most domain acquisition systems in the semantic web community model each domain separately and restrict the system to a single language or a limited set of languages. Furthermore, they require knowledge engineers and language technology experts to do the modeling. The KYOTO system on the other hand is specifically designed to develop and maintain crosslinguistic and crosscultural consensus on the meaning of relevant terminology. Thus, KYOTO is an open system that can be extended and curated directly by the users without requiring knowledge engineering or language technology skills.

The system shares many features with the Wikipedia in that it allows user groups to agree on the interpretation and meaning of relevant concepts. Crucially, the terms’ definitions are formalized so that computer programs can use them to mine the texts selected for analysis by the groups’ members. The knowledge acquisition process furthermore includes automatic term mining from the user-provided text documents. A specially designed editing environment guides the users in selecting and defining relevant terms while masking the formal knowledge structures underlying the definitions. In this way, domain-specific lexical resources in the form of wordnets are being created by the users with support from the system. All wordnets are linked to the Princeton English WordNet.

In order to further formalize the semantic representation of the selected terms and to make them universally accessible, the editor prompts the users to encode appropriate formal constraints and semantic relations. Here, too, complex knowledge structures are hidden. This editing step results in a uniform domain ontology created by different user groups.

The representation of the domain terms in both the wordnets and the shared ontology is anchored to generic wordnets and a generic central ontology.

3 Knowledge Representation in KYOTO

KYOTO uses a model in which knowledge is distributed over three distinct but interrelated layers: domain terminologies, generic and domain wordnets and a central ontology. Terms from domain-specific resources or acquired from documents are mapped onto wordnets in the different languages, which in turn are mapped to a central ontology. The mappings across these resources can be partial, so long as it is possible to reach a matching concept through internal inheritance relations.

Fundamental and frequent concepts are mapped from the seven KYOTO wordnets to the central ontology, which in turn facilitates the mappings for domain-specific concepts done by the experts. Such concepts—mostly lexicalized by nouns–are edited through the Wiki (Wikyoto) platform with the help of questionnaires, or “interviews”, that probe for ontologically important distinctions. Generated from the definitions of the terms, the interviews consist of simple simple yes-no-questions posed to the expert editor.

We focus in particular on the distinction between types and roles. Rigidity is an ontological property that characterizes the former but not the latter. For example, “dog” is a type, but “pet” is a role. A given dog will always be a dog, but its pethood may be a transitory state that ceases when the dog is abandoned by its owner. Similarly, an endemic species may cease to be endemic when conditions in its native habitat change. In the specific domain that KYOTO focuses on, non-rigid concepts are particularly pervasise, as the relevant documents tend to discuss and report trends and observations of processes and states. Hicks and Herold (in press) examine the effectiveness of lexical-semantic patterns like *Xs and other Ys* and *X used to be a Y* that distinguish types from roles within and across languages.

KYOTO’s three-layered model makes it possible to map massive amounts of data from domain- and language-specific wordnets to the generic wordnets and to the central, formal ontology with its rich axioms for modeling processes and qualities. Representing knowledge in the wordnets takes advantage of the inheritance (hyponymy) relation, which relates specific to more general concepts. This allows for an efficient and redundancy-free encoding. The ontology is deliberately kept small and sparse. It focuses on the presentation of a limited set of disjoint types and on the processes and states which these types are typically a part of. Distributing the knowledge between ontology, generic and domain wordnets achieves the optimal balance between expressiveness and recall of knowledge and faciliates automatic inferencing over entities and events.

Acknowledgment

The work described here is funded by the European Union (7th Framework) contract ICT-211,423 and was carried out by the members of the consortium under the direction of the project coordinator Piek Vossen.

References

1. Bosma, W., Vossen, P.: Bootstrapping language neutral term extraction. In: Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010, Malta, May 17-23 (2010)
2. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
3. Hicks, A., Herold, A.: Cross-lingual Evaluation of Ontologies with Rudify. In: Fred, A., Dietz, J.L.G., Liu, K., Fillipe, J. (eds.) Knowledge Discovery, Knowledge Engineering and Knowledge Management. Communications in Computer and Information Science. Springer, Heidelberg (in press)

4. Vossen, P., Bosma, W., Cuadros, M., Rigau, G.: Integrating a large domain ontology of species into WordNet. In: Proceedings of the 7th International Conference on Language Resources and Evaluation LREC 2010, Malta, May 17-23 (2010)
5. Vossen, P., Agirre, E., Calzolari, N., Fellbaum, C., Hsieh, S., Huang, C., Isahara, H., Kanzaki, K., Marchetti, A., Monachini, M., Neri, F., Raffaelli, R., Rigau, G., Tescon, M.: KYOTO: A system for Mining, Structuring and Distributing Knowledge Across Languages and Cultures. In: Proceedings of LREC 2008, Marrakech, Morocco, May 28-30 (2008)
6. Vossen, P., Agirre, E., Bond, F., Bosma, W., Fellbaum, C., Hicks, A., Hsieh, S., Isahara, H., Huang, C., Kanzaki, K., Marchetti, A., Rigau, G., Ronzano, F., Segers, R., Tesconi, M.: KYOTO: a Wiki for Establishing Semantic Interoperability for Knowledge Sharing across Languages and Cultures. In: Blanchard, E., Allard, D. (eds.) Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models. IGI (in press)

Evolution of the ASR Decoder Design

Miroslav Novak

IBM T.J. Watson Research Center, Yorktown Heights, 10598 NY, USA
miroslav@us.ibm.com
https://researcher.ibm.com/researcher/view.php?person=us-miroslav

Abstract. The ASR decoder is one of the fundamental components of an ASR system and has been evolving over the years to address the increasing demands for larger domains as well as the availability of more powerful hardware. Though the basic search algorithm (i.e. Viterbi search) is relatively simple, implementing a decoder which can handle hundreds of thousands of words in the active vocabulary and hundreds of millions of n-grams in the language model in real time is no simple task. With the emergence of embedded platforms, some of the design concepts used in the past to cope with limitations of the available hardware can become relevant again, where such limitations are similar to those of workstations of early days of ASR. In this paper we will describe various basic design concepts encountered in various decoder implementations, with the focus on those which are relevant today among the fairly large spectrum of available hardware platforms.

1 Introduction

The ASR recognizer implements the fundamental equation of the speech recognition [1]:

$$\boldsymbol{W}' = \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}|\boldsymbol{O}) \tag{1}$$

$$= \arg\max_{\boldsymbol{W}} \frac{P(\boldsymbol{O}|\boldsymbol{W})P(\boldsymbol{W})}{P(\boldsymbol{O})} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{O}|\boldsymbol{W})P(\boldsymbol{W}) \tag{2}$$

$$= \arg\max_{\boldsymbol{S}} \prod_{t=0}^{T-1} P(\boldsymbol{o}_t|s_t)P(s_{t+1}|s_t) \tag{3}$$

where $\boldsymbol{W} = [w_1 \dots w_k]$ is a sequence of words and $\boldsymbol{O} = [o_1 \dots o_T]$ is a sequence of feature vectors representing the utterance. The role of the decoder is to find the word sequence which maximizes the probability $P(\boldsymbol{W}|\boldsymbol{O})$. Using the Bayes' rule, this probability can be factored into the Acoustic Model (AM) probability $P(\boldsymbol{O}|\boldsymbol{W})$ and the Language Model (LM) probability $P(\boldsymbol{W})$. Since $P(\boldsymbol{O})$ is not needed to find the maximum, it can be omitted. But this omission has an important consequence: the likelihood of the best path $P(\boldsymbol{O}|\boldsymbol{W})P(\boldsymbol{W})$ no longer represents the probability of the utterance. This means that it cannot be reliably used to determine if the best word sequence within a specific LM is a good representation of the utterance. For example, to be able to reject utterances which are not legal for a given LM, some approximation of $P(\boldsymbol{O})$ must be used in the confidence measure computation.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 10–17, 2010.

The equation (3) suggests how to perform the search using the Hidden Markov Models (HMM) framework. All the knowledge sources (i.e. AM and LM) are combined into single HMM network, in which each word is represented by a sequence of states with the emission distribution $P(\boldsymbol{o}_t|s_i)$. Ideally, each state should have a unique distribution representing a particular part of the word in various acoustic contexts to model the co-articulation correctly. Since the amount of the training data is limited, the distributions are shared among the words and contexts. In a typical ASR system, the distribution of a single state models one third of a phoneme (divided into start, middle and end section) in a context of one or two phonemes on each side. The states are clustered in order to provide enough training samples for each unique distribution. Decision trees asking question about the surrounding phonetic context are commonly used [2]. Their advantage in comparison to other methods (e.g. bottom up clustering) is that they provide a valid state for a context unseen in the training data. The state distribution is then modeled with a Gaussian Mixture Models (GMM), often restricted to have a diagonal covariance matrix.

The size of the phonetic context is one of the most critical factors in the decoder design. In the early days of ASR, the word-internal context modeling was prevalent, i.e. the co-articulation effect of the phonemes across word boundaries was ignored. This approach significantly simplifies the HMM network construction, since the state sequence for each word can be easily constructed without the cross-word constraints. This approach can still be found today on some low resource embedded platforms or as a first-pass model in multi-pass decoders. It also results in less costly lattice generation and higher decoding speed. The word error rate increases by 15–30% in comparison to full cross word modeling.

As the next step in the complexity, the left cross-word context is often used as a good trade-off between the accuracy and complexity. Its impact on the word error rate increase is usually less than 10%. The HMM graph for each word needs to have multiple entry points for each possible predecessor context (fan-in) and all the predecessors are know when the graph is built from left to right (e.g. when built dynamically during the search).

The right cross-word context is by far the most complex technique, in addition to fan-in, each word requires multiple end points for all possible successors (fan-out), which may not be known when the graph is constructed dynamically. This modeling method has become more widely used with adoption of *Weighted Finite State Transducers* (WFST) methodology for the graph construction.

The AM provides $P(\boldsymbol{o}_t|s_i)$, i.e. the probability of a state s_i generating an observation $\boldsymbol{o}_t$. The LM probabilities applied to transitions between words and HMM state transition probabilities within each word are combined in $P(s_{t+1}|s_t)$ and often the state transitions (modeling the state durations) are omitted. The decoder task is then to find the best path $\boldsymbol{S} = [s_1 \dots s_T]$ through an HMM network. The decoder design must to address the following issues:

- the composition of combining the knowledge sources may be costly (in term of the memory and CPU use),
- the result of this composition may be too large to fit into memory,
- the knowledge sources may change frequently (e.g. frequent vocabulary updates),

- the application may require more information from the decoder that just the best path (e.g. word lattice).

Since the power of the hardware has evolved during the period the statistical speech recognition has been studied, some of the issues became less critical than others. At the same time the amount of the available training data has increased significantly, resulting in much larger AMs and LMs with much larger vocabularies. Also, the effort to implement the ASR system on a variety of hardware platforms (from cell phones to large servers) may bring some methods used in the early days of speech recognition into our attention again.

2 Asynchronous Search

There are two main approaches to the search implementation, asynchronous (also knows as stack search) and synchronous (Viterbi).

The asynchronous search has become significantly less popular after the wide spread use of WFSTs. Its main advantage is memory efficiency. It is often called a stack search, because it derived from the A^* algorithm. The A^* algorithm uses a stack to rank hypotheses (incomplete word sequences $\boldsymbol{W}_{1,k}$) according to estimates of the likelihood $\bar{P}(\boldsymbol{W}_{1,k}|\boldsymbol{O}) = P(W_{1,k}|\boldsymbol{O}_{1,t})e(\boldsymbol{O}_{t+1,T})$ where $e(\boldsymbol{O}_{t+1,T})$ is an estimate of the likelihood of matching the remaining portion of a complete hypothesis of the unseen segment of the utterance. When A^* is used for tasks when $e(\boldsymbol{O}_{t+1,T})$ is equal to the likelihood of the best path over the utterance $\boldsymbol{O}_{t+1,T}$ (e.g.. in lattice rescoring or N-best generation), the search is admissible (it will find the best path) and efficient (the first complete path found is the best one).

In the *envelope* search [3], the estimate $e(\boldsymbol{O}_{t+1,T})$ is computed as a distance from the best matching path at the time t. Since the condition for admissibility and efficiency is not satisfied, all paths falling with a certain likelihood difference Δ from the best path need to be extended, the size of Δ is used to balance efficiency and admissibility. The search itself is organized as a tree search, where each branch represents a single word, each iteration one incomplete path is extended by a set of possible following words, creating a set of new paths. Each new path is then inserted into the stack if its likelihood falls within the Δ distance. A *Fast Match* [4] is usually used to limit the number of extended words in each iteration to several hundreds.

The tree structure keeps track of the word hypotheses quite efficiently, since a relatively small amount of information is stored for each word extension. The memory needed to compute the likelihood of each word extension (i.e. to evaluate the HMM trellis, this step is also called *Detailed Match*) can be recycled from iteration to iteration. Furthermore, the Detailed Match likelihoods can be reused under certain conditions for extension of other paths that end at the same time (when they represent the same acoustic but different LM contexts).

A major disadvantage of the asynchronous search is its algorithmic complexity with many tuning parameters. The envelope search relies on a proper determination of the word end, which can become unreliable in noisy conditions, so at low levels of SNR, the Viterbi decoder often performs better.

3 Viterbi Search

The Viterbi search [5] owes its popularity to its simplicity. Most of the complexity in today's recognizers comes not from the search implementation itself, but from the HMM graph construction and generation of additional information such as phone or word lattices.

In the early days of ASR, tasks were limited to word loops of tens or at most hundreds of words. The graph construction for such tasks is straightforward, especially when no cross-word context modeling is used, by simply replacing each word with its corresponding state sequence. Word-pair grammars were used to incorporate the LM by introduction of non-emitting connections between each word initial and final states. But this method is not scalable to larger vocabulary sizes, since the number of connections grows quadratically. The introduction of the *backoff* LM [6] was crucial for the advancement of the Viterbi decoder, since it can be represented (with some small approximations) as a WFST. The cross word connections are instantiated only for cases when the corresponding n-gram exist, otherwise the connections are collapsed to a lower order n-gram.

With the increasing vocabulary sizes and the n-gram order, the construction of the whole graph became unfeasible. Several techniques based on lexical trees were developed [7]. The lexical tree is formed by merging common state prefixes for all words in the vocabulary. This is very beneficial for decoding speed, but a single tree can model a unigram LM only. Two main approaches to keep the track of unique histories can be found. One possibility is to create a new copy for each new LM history. New histories are created at the leaves of these trees where a word label is reached and the equivalent histories are merged as soon as the n-gram constraints are satisfied. Pruning thresholds are applied to limit the number of active trees. The second method uses a single copy of the tree but multiple tokens are propagated during the search to represent multiple histories. The later methods seems to be more widely used, one reason may be that it is easier to merge the LM histories before the leaves are reached, which results in word tail sharing.

The concept of prefix and tail sharing was formalized when WFSTs were introduced into speech recognition [8]. WFSTs provide a solid theoretical framework for the operations needed for decoding graph construction. A decoding graph is the result of a composition

$$S = HC \circ L \circ G, \tag{4}$$

where G represents a language model, L represents a pronunciation dictionary and HC translates phone sequences to context dependent HMM states. The determinization (prefix sharing) and minimization (tail sharing) produce minimal graphs leading to the most efficient decoder implementations. This method is applicable to the incorporation of any knowledge source that can be expressed as a WFST.

The minimization effect is particularly pronounced when right cross-word modeling is used. The label pushing step moves the word labels before the fan-out portion the state graph, so the fan-outs can be shared among multiple word, which leads to a significant reduction in the graph size. Further size reduction can be achieved by application of global minimization (which cannot be easily performed in a dynamic scheme).

There are several caveats though. Construction of HC is not trivial, especially for a wide cross word context [9,10]. The composition can be quite time and memory consuming and the resulting graph may be too large for practical use. For these reasons, *on-the-fly* variations of the composition are popular [11], where the composition is split into two steps. In the first, static step, $HC \circ L$ is constructed an minimized, in the second step, performed at runtime, the final composition with G is performed for only a small portion of the graph corresponding to active states. This leads to a significant memory saving for the price of a higher CPU cost during the runtime.

The minimization produces a graph with a minimal number of states. Unfortunately decoding cost is more related to the number of arcs in the graph. There is no known algorithm for arc minimization. In fact, in some situations the determinization (a required step before the minimization) can increase the number of arcs significantly. An example is G representing a backoff LM. Epsilon transition (i.e. without a label) are used there to find probabilities for n-grams unseen in the training data using lower order n-grams. Determinization (which includes epsilon removal) would produce a graph with $|V|^n$, which is intractable even for small vocabularies V. For this reason, the epsilon arcs in a backoff LM are always treated as a part of the vocabulary and never removed.

Conceptually similar but implementation-wise a very different approach to address the static graph size is based on two-pass decoding. In the first pass a word lattice is generated using a smaller LM, which is then rescored in the second pass using the full model. It is usually desirable to use a model as large as possible in the first pass to reduce the probability of search errors. Contrary to the intuition, decoding of a large graph is often faster, because it represents a more accurate model and the pruning can be done more aggressively. The rescoring approach is also very practical when complex LMs are used, which cannot be easily represented by a WFST, such as adaptively interpolated LMs, class based LMs etc.

When comparing these three methods (i.e. full static single-pass, dynamic single-pass and two-pass lattice rescoring) it is difficult to assess the superiority of one method over the others. The performance depends on many aspects, including the size of the task and the targeted platform. For example the speed advantage of the static decoder may be diminished when it is implemented on a platform with a slow access to the main memory with a high cache miss rate. Similarly, the memory advantage of the dynamic approach is reduced in a multi-core environment when multiple data streams are processed using a single model located in shared memory.

The concept of the Fast Match can be used in the Vitebi search as well, but it is less common than in the asynchronous search. For example word loop with a simplified AM can be used to find possible word begin times. Because the Viterbi search produces discrete time boundaries between words rather than a time interval, some heuristic must be applied to utilize the FM results efficiently, which can make the decoder less robust.

4 Language Model

Regardless of the decoder implementation, the LMs used by ASR systems can be divided into two major groups: grammars and n-grams. While both models can be represented as a WFST, they are different in many other aspects. Grammars are rules based, i.e. written by a domain expert in some form of a regular language and then compiled into an WFST representation. They can be stochastic, i.e. with weights assigned to each transition by either the domain expert or by training them on observed data. Well designed grammars tend to yield better accuracy and speed, since they can significantly restrict the search space (low perplexity). Grammars used to be a clear choice in early ASR systems and are still heavily used by embedded systems. As the vocabulary sizes grew with more advances system, n-grams became more popular.

Grammars perform well when all utterances are in domain, i.e. the grammar matches all user utterances exactly. Design of such grammar gets harder as the vocabulary size and task complexity grows. It is difficult for the designer to predict all possible ways the user will want to interact with the system as well as it is for the user learn and remember all legal sequences. The n-gram based system (often followed by an NLU processing step) does not have this limitation.

While grammars have good in-domain performance they often have very poor out-of-domain performance, i.e. they are not very good at recognizing situations when the user says something outside of the grammar space. In many cases the decoder will come up with the closest matching legal phrase and a reliable confidence score must be computed to verify the utterance. The advantage of using an n-gram model is that it provides a "natural" background model by covering the search space much more uniformly than the grammar. This leads to more reliable confidence scores or an option to reject the phrase by the NLU unit.

There is an interesting class of applications characterized by a large vocabulary size and high complexity but low perplexity. A typical example is search of names and addresses. Having one large grammar driving the search may not be practical due to the size of such grammar and probability of search errors. It may be more practical to factor the grammar and recover the lost dependency information (i.e. matching specific name with specific address) at later processing step. For this scenario, the decoder must provide a reliable list of alternatives for each factor, either as a word lattice or as a n-best list.

A possible disadvantage of n-grams is that they require a lot of training data. Acquisition of such data may be expensive, especially for applications in new domains.

Decoders which rely on the use of Fast Match tend to perform worse when grammars are used, because at some states of the grammar the Fast Match is unnecessary and introduces more search errors.

To combine benefits of n-grams and grammars, hybrid systems can be utilized. The LM is factored into a general part modeled by an n-gram and several specific parts represented be grammar. This can be seen as an extension of a class based language model, where instead of word classes (e.g. days of a week) more complex grammars are used. The language model scores inside each grammar factor is provided by this grammar only and the word sequence segment matching the grammar represents a single token in the n-gram model history.

This factorization has several advantages:

1. the n-gram model can be trained independently of the grammars, the amount of the training data needed is reduced,

2. the grammars can be modified without retraining of the model, possibly at runtime (late binding) [12],
3. alignment information can provide important information for the following NLU processing step.

5 Lattice Generation

Phone or word lattice are used in many situations, such as N-best generation for multiple choices in the GUI, rescoring with complex knowledge sources, confidence measure computation, discriminative training etc. The standard Viterbi algorithm needs to be modified, because it is necessary to keep track of all possible competing paths. This can be very memory demanding so some form of approximation is usually needed. The following are examples of some of the approaches.

- Keeping track of the competing paths only for states with two or more arcs coming from other states (i.e. not counting the self loops). The limitation of this method is that it can only find alternatives at the time boundaries introduced by the best path (or recursively by the alternatives found).
- Propagation of multiple tokens with trace records created for each token at the word ends. This method leads to some of the most precise lattices, but it has a runtime overhead [13].
- Keeping track of word ends only and using LM or context constraints to create lattice links. The disadvantage of this method is that can produce non-existing alternatives [14].

6 Conclusion

We have briefly described several methods which have been used in the implementation of ASR systems. This is not an exhaustive list, there are many more interesting methods and details which can be found in the literature.

References

1. Bahl, L.R., Jelinek, F., Mercer, R.L.: A maximum likelihood approach to continuous speech recognition. IEEE Transactions on Pattern Analysis and Machine Inteligence 5(2), 179–190 (1983)
2. Bahl, L.R., De Souza, P.V., Gopalakrishnan, P.S., Nahamoo, D., Picheny, M.A.: Context dependent modelling of phones in continuous speech using decision trees. In: Proceedings DARPA Speech and Natural Language Processing Workshop, pp. 264–270 (1991)
3. Gopalakrishnan, P.S., Bahl, L.R., Mercer, R.L.: A tree search strategy for large vocabulary continuous speech recognition. In: Proc. ICASSP 1995, May 1995, pp. 572–575 (1995)
4. Bahl, L.R., De Gennaro, S.V., Gopalakrishnan, P.S., Mercer, R.L.: A fast approximate acoustic match for large vocabulary speech recognition. IEEE Transactions on Speech and Audio Processing 1(1), 59–67 (1993)
5. Forney Jr., G.D.: The Viterbi algorithm. Proceedings of the IEEE 61, 268–278 (1973)

6. Katz, S.M.: Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoust., Speech and Signal Processing 35(3), 400–401 (1987)
7. Ortmanns, S., Eiden, A., Ney, H.: Improved lexical tree search for large vocabulary speech recognition. In: Proceedings of ICASSP 1998, vol. 2, pp. 817–820 (1998)
8. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. Computer Speech & Language 16(1), 69–88 (2002)
9. Chen, A.: Compiling large-context phonetic decision trees into finite-state transducers, Geneva, Switzerland, pp. 1169–1172 (2003)
10. Novak, M.: Incremental composition of static decoding graphs. In: Proceedings of Eurospeech 2009, Brighton, UK (2009)
11. Caseiro, D., Trancose, I.: A specialized on-the-fly algorithm for lexicon and language model composition. IEEE Transactions on Audio, Speech and Language Processing 14(4), 1281–1291 (2006)
12. Schalkwyk, J., Hetherington, L., Story, E.: Speech recognition with dynamic grammars using finite-state transducers. In: Proc. of Eurospeech 2003, pp. 1969–1972 (2003)
13. Saon, G., Povey, D., Zweig, G.: Anatomy of an extremely fast lvcsr decoder. In: Proceedings of Interspeech 2005, pp. 549–552 (2005)
14. Novak, M.: Memory efficient approximative lattice generation for grammar based decoding. In: Proceedings of Eurospeech 2005, Lisbon, Portugal (2005)

Part II

Text

“**Text:** a book or other written or printed work, regarded in terms of its content rather than its physical form: *a text which explores pain and grief.*”
NODE (The New Oxford Dictionary of English), Oxford, OUP, 1998, page 1998, meaning 1.

Encoding Event and Argument Structures in Wordnets

Raquel Amaro, Sara Mendes, and Palmira Marrafa

CLG, Center of Linguistics of the University of Lisbon,
Av. Prof. Gama Pinto, 2, 1649-003 Lisbon, Portugal
{ramaro,sara.mendes}@clul.ul.pt, palmira.marrafa@netcabo.pt
http://www.clul.ul.pt/clg

Abstract. In this paper we propose the codification of argument and event structures in wordnets, providing information on selection properties, semantic incorporation phenomena and internal properties of events, in what we claim to be an affordable procedure. We propose an explicit expression of argument structure, including default and shadow arguments, through three new relations and a new order feature. As synsets in wordnets are associated to a given POS, information on the selection properties of lexical items is added. We show that the systematic encoding of event structure information, through five new features at synset level, besides providing the grounds for describing the order of arguments, enriches the descriptive power of these resources. In doing so, we crucially contribute to making wordnets rich and structured repositories of lexical semantic information, that allow for the extraction of argument and event structures of lexical items, thus enhancing their usability in NLP systems.

Keywords: Relational lexica, wordnet, argument structure, event structure.

1 Introduction

Wordnets are electronic relational databases structured as networks of relations between synsets (sets of synonymous word forms of the same POS), and focusing on conceptual and semantic relations such as synonymy, antonymy, hyperonymy, meronymy, and so on. The original model corresponds to the Princeton WordNet [1,2], a lexical-conceptual database for English containing nouns, verbs, adjectives, and adverbs.

The use of WordNet as a lexical base in NLP applications has made its shortcomings apparent in the perspective of application developers and has led to the need for finer-grained lexical descriptions allowing computational systems to deal automatically with various complex linguistic phenomena in a general and systematic way. For these reasons, and in order to improve the usability of this resource for a variety of applications, the association of semantic and some syntactic information to the WordNet model has been object of research since its appearance (see [3,4,5] or [6], among others).

In this paper we propose the encoding of argument and event structures in wordnets, providing information on selection properties, semantic incorporation phenomena and internal properties of events, in what we claim to be an affordable procedure. Aiming at building a computational relational lexicon that models both the semantic and syntactic properties of lexical items, our proposal consists in explicitly expressing argument and event structures in WordNet.PT [7], an electronic relational database for Portuguese,

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 21–28, 2010.

developed following the EuroWordNet framework, henceforth EWN [8], including default and shadow arguments (as defined in the Generative Lexicon model, henceforth GL [9,10]), through a small set of new relations and features which entirely preserve the architechture of the model.

1.1 Related Work

The association of semantic and some syntactic information to the WordNet model has been the subject of much research work developed since its appearance (see for instance [3,4] or [5]). More recently, and with regard to cross-POS relations concerning selectional properties, several approaches have been taken. [6], for instance, propose the integration of selectional properties in WordNet through the extension of existing word-to-class statistical models to class-to-class preferences that allow computational systems to learn the selectional preferences for classes of verbs, and associate statistic information on the selectional preferences of each sense of a given word (synset). [11], in a similar approach, focus on the identification and integration of phrasets (a type of synset containing multi-word units) to model combinatory idiosyncrasies of lexical units; when the introduction of a phraset is not justified, syntagmatic relations between verbs and their arguments are stated. [12] propose the introduction of instances of verb subjects and direct objects, extracted from linguistically analyzed and annotated corpora. On a different approach, and aiming at enhancing the density of the network, [13] propose the introduction of a new type of relation, based on the concept of evocation, to connect synsets which evoke or bring other synsets to mind.

One of the major results of research on WordNet enhancement is EWN [8], a resource reflecting research both on lexical semantics and on the usability of wordnets in NLP applications. Such research resulted in the definition and implementation of a wider variety of lexical-conceptual relations than the set used in Princeton WordNet, focusing on more comprehensive lexical-conceptual relations and cross-POS relations. Specifically, the EWN model describes selectional properties through role relations, as the ones illustrated below, in the sense that these establish a relation between event-denoting synsets and synsets denoting the participants typically involved in them.

1. a. ROLE AGENT relation: *an entity denoted by N1 is the one/that who/which does the event denoted by V1/N2, typically in intentional way*
   b. ROLE PATIENT relation: *an entity denoted by N1 is the one/that who/which undergoes the event denoted by V1/N2*

Role relations are based on thematic role assignment, thus somehow assuming the function of syntactic mapping: agents are commonly assumed to be syntactically realized in subject position, patients in object position, and so forth, as shown by the EWN tests presented above. These relations, however, are defined to connect nodes that lexicalize a given thematic function of an event, or whose thematic function is necessarily or typically associated to the concept denoted, and do not explicitly express selectional properties. Also, as most relations in wordnets, role relations are designed to represent semantic and conceptual properties of the concepts lexicalized in a given language, which do not necessarily reflect selectional properties with precision.

2. a. {*teacher*}N ROLE AGENT {*teach*}V/{*teach*}V INVOLVED AGENT {*teacher*}N, **but** *teach* selects an animated entity as subject.

Focusing on the EWN model, [14] and [15] argue that wordnets, being concept-based computational lexica, should include information on event and argument structures for allowing computational grammars to cope with a number of different lexical semantics phenomena. Specifically, these authors propose new cross-POS relations linking adjectives and nouns, but also the representation of telicity of LCS deficitary verbs, directly related to the structure of these events, through the inclusion of a telic sub-event relation. Also, the relevance of verb argument structure for determining co-hyponymy compatibility is shown and the need to integrate prepositions in wordnets motivated.

Following this approach, in this paper we propose the encoding of argument and event structures in wordnets, through the definition of a small set of new relations and features, providing information on selectional and event internal properties of lexical items, as well as order constraints on their syntactic mapping, without compromising the WordNet model.

2 Encoding Argument Structure

As described above, the EWN model encodes some selection properties of lexical items through *role* relations, that connect nodes lexicalizing a specific thematic function of a given event or whose thematic function is necessarily or typically associated to that event. However, and in spite of being based on thematic role assignment, these relations do not necessarily encode the argument structure of lexical items nor explicitly express selection properties.

In GL ([9,10]), the argument structure is a level of representation in which the number and type of arguments of a lexical item is stated, including the definition of the semantic properties of its logical arguments, but also syntactic mapping information. The integration of this information in wordnets results in an increase of relevant information available on the semantic and syntactic properties of lexical items. The information defined in the argument structure significantly complements the lexical-semantic information represented through *role* relations and contributes to a more accurate and complete description of the data, enhancing the usability of wordnets as a resource for meaning computation purposes.

Our proposal for encoding argument structure consists in explicitly expressing it through three new relations and a new order feature, overcoming the shortcomings of establishing correspondences between thematic role assignment and syntactic mapping, and providing some information on subcategorization properties. These three relations reflect the three types of arguments considered in GL – true arguments, shadow arguments and default arguments – and are informally defined as follows:

3. SELECTS/ IS SELECTED BY relation (for true arguments)
{synset}1 SELECTS {synset}2 and {synset}2 IS SELECTED BY {synset}1
iff ∃ x: $x \in$ {synset}1 and ∃ y: $y \in$ {synset}2, and the syntactic realization of x requires the syntactic realization of y, or of z, if ∃ z: $z \in$ {synset}3 hyponym of {synset}2.

Example: {*gallop*}$_{V}$ SELECTS 1 {*equine*}$_{N}$

4. INCORPORATES/IS INCORPORATED IN relation (for shadow arguments)
   {synset}$_1$ INCORPORATES {synset}$_2$ and {synset}$_2$ IS INCORPORATED IN {synset}$_1$ iff:
   i) the concept denoted by the {synset}$_1$ entails the specific concept lexicalized by the {synset}$_2$;
   ii) $\exists x$: $x \in$ {synset}$_1$ and $\exists y$: $y \in$ {synset}$_2$, and the co-occurrence of x and y is only licensed by subtyping or specification processes; and
   iii) in case of conjoint incorporations, ii) only applies to the element with reference potential.
   Example: {*poison*}$_{V}$ INCORPORATES 3 {*with*}$_{Prep}$ conj1:1
   INCORPORATES 3 {*poison*}$_{N}$ conj2:1
5. SELECTS BY DEFAULT/IS SELECTED BY DEFAULT BY relation (for default arguments)
   {synset}$_1$ SELECTS BY DEFAULT {synset}$_2$ and {synset}$_2$ IS SELECTED BY DEFAULT {synset}$_1$ iff:
   i) the concept denoted by the {synset}$_1$ entails the underspecified concept denoted by the {synset}$_2$;
   ii) $\exists x$: $x \in$ {synset}$_1$ and $\exists y$: $y \in$ {synset}$_2$ and the co-occurrence of x and y is only licensed by subtyping or specification processes; and
   iii) in case of conjoint default selections, ii) only applies to the element with reference potential.
   Example: {*build*}$_{V}$ SELECTS BY DEFAULT 3 {*of*}$_{Prep}$ conj1:1
   SELECTS BY DEFAULT 3 {*material*}$_{N}$ conj2:1

In order to index the arguments established by SELECTS, INCORPORATES and SELECTS BY DEFAULT relations to a given order, it is necessary to implement an order feature, expressed by numerical tags. Arguments are integrated in a list ⟨1, 2, ..., n⟩, ordered from the less oblique to the more oblique[1].

The combination of the SELECTS relation with the numerical tags of the order feature allows for extracting the order in which arguments are syntactically realized, depending on their position in the list defined at event structure level. The relation tagged with 1 indicates the argument that is realized in the less oblique position (subject position, in the case of verbs, object position, in the case of nouns), the relation tagged with 2 indicates the argument that is realized in object position, and so on. The INCORPORATES and SELECTS BY DEFAULT relations, although referring to arguments generally not syntactically expressed, are also tagged to assure their correct syntactic position in subtyping or specification contexts, as showed bellow:

6. a. #[The man]$_1$ poisoned [the cattle]$_2$ [with poison]$_3$.
   b.[The man]$_1$ poisoned [the cattle]$_2$ [with arsenic]$_3$.

[1] Order here refers to the so-called basic order of constituents. The list of arguments provides the basic unmarked position of arguments, not aiming at accounting for other possible syntactic positions.

7. a. #[The man]$_1$ built [the house]$_2$ [of material]$_3$.
   b. [The man]$_1$ built [the house]$_2$ [of wood]$_3$.

Although the integration of argument information in wordnets adds some effort to the encoding task, it does not amount to a true surplus of work since argument structure properties are directly related to the concepts denoted and are frequently required to disambiguate senses and/or to determine the degree of sense differentiation. Thus, the additional work required can amount to significant gains in coherence and sustained sense differentiation options, which make it worth it.

3 Integrating Event Structure

Event structure is the level of representation that regards the internal properties of an event associated to a lexical item. In GL, this level of representation refers to four internal properties of events: their subevents list (E_1= $\mathbf{e_1}$,. . . , E_n= $\mathbf{e_n}$); their Aktionsart type; temporal and order restrictions of their subevents; and their head subevent.

Event structure is the most internal level of representation of event denoting lexical items in GL, in the sense that it comprises semantic properties that are not necessarily (or even not at all) related to external elements. For these reasons, and in contrast with argument structure information, event structure cannot be integrated in wordnets via lexical-conceptual relations established between existing synsets, since the properties it defines are hardly ever lexicalized and thus are not reflected in the nodes in the network.

Given these specificities, we propose to encode event structure as additional information at the synset level, through the use of features that mirror the aforementioned attributes. Also, we claim the need for introducing a new feature that enables the statement of the list of arguments of a given event.

The features presented in Table 1 mirror the attributes used in the GL model. The attribute *event type* can have one of three possible values, corresponding to the three event types, as defined in [16]: **state** (atomic event, not evaluated with regard to any other), **process** (sequence of identical events (complex or not)) and **transition** (event evaluated regarding another event, composed of a process that culminates in a final state, different of the initial one).

The value of the attribute *arguments* consists in the list of arguments selected by the event denoted by the synset, ordered from the less oblique to the most oblique one.

Table 1. Event structure features

ATTRIBUTE	VALUES
event type	*state* *process* *transition*
arguments	$\langle$ *1, 2, . . . , n* $\rangle$
subevents	$e_1(2,3), \ldots, e_n(1,2)$
restrictions	$< \alpha$, $\circ\alpha$, $< \circ\alpha$
head	$e_{1 \ldots n}$

The natural numbers making up the elements of this list correspond to the *order* feature values (cf. Section 2) associated to argument structure relations, allowing the indexation of the selected nodes to a given position in the list.

The *subevents* attribute allows for listing the subevents that compose transition denoting events (according to the established typology of events), with information on the arguments of each subevent. For instance, a transition type event, such as the one denoted by *build*, has as subevents an event argument that corresponds to the process (of building) that leads to a final state (e_1(1,2,3)) and a second event argument that corresponds to this final state (of being built) (e_2(2,3)).

The *restrictions* attribute allows for expressing the three possible temporal ordering relations of subevents established in the GL model: exhaustive ordered part of ($< \alpha$), exhaustive overlap part of ($\circ\alpha$), and exhaustive ordered overlap ($< \circ\alpha$).

Finally, the *head* feature determines the head subevent of a given event-denoting lexical item, this way accounting for Aktionsart properties (achievement vs. accomplishment type events), as well as for events lexically underspecified with regard to event headedness, that typically enter causative/inchoative alternation constructions (see [10], for a detailed discussion).

The features presented in Table 1 allow the expression of event structure without any loss of information. Note, however, that lexicalized subevents are also stated through lexical-conceptual relations at the network level. Thus, verbs that have conceptually individuated and lexicalized subevents, such as {*breathe*} or {*sadden*}, are respectively characterized through HAS SUB-EVENT and HAS TELIC SUBEVENT relations, as follows:

8. {*breathe*}HAS SUBEVENT {*inhale*};{*breathe*} HAS SUBEVENT {*exhale*}
9. {*sadden*}V HAS TELIC SUBEVENT {*sad*}ADJ

Along the lines of what has been argued in the previous section, we consider adding information on event structure to wordnets to be also an affordable effort in wordnet development, involving only filling in the values of new features associated to event denoting synsets.

Although features convey additional information that is not expressed through lexical-conceptual relations, our motivation is that the systematic statement of event structure information, besides providing the grounds for argument order description and consequent syntactic mapping, enriches the descriptive power of these resources, crucially contributing to making wordnets rich and structured repositories of lexical semantic information, thus allowing the extraction of argument and event structures of lexical items, and hence fine-grained and rich lexical entries.

4 Final Remarks

The enhancement strategies proposed in this paper contribute to a significant enrichment of wordnets with semantic and syntactic information that, along with the available lexical structures, the lexical-conceptual relations that connect them in relational models of the lexicon and the percolation of information intrinsic to this model of the lexicon, render wordnets more complete and usable resources for a great variety of NLP

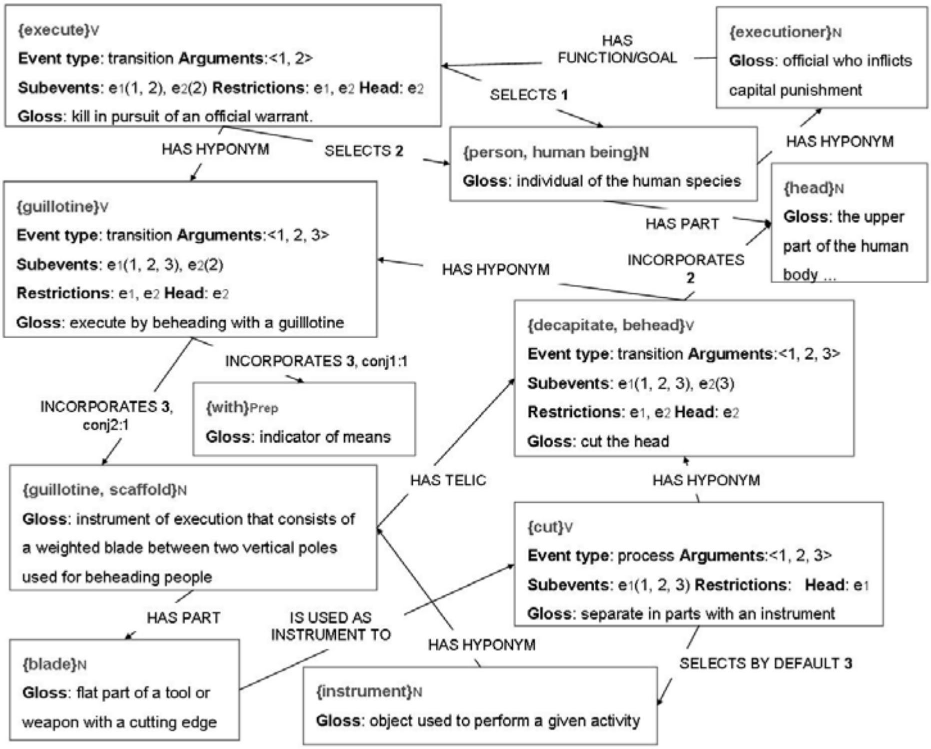

{execute}V
Event type: transition Arguments:<1, 2>
Subevents: e1(1, 2), e2(2) Restrictions: e1, e2 Head: e2
Gloss: kill in pursuit of an official warrant.
HAS FUNCTION/GOAL
{executioner}N
Gloss: official who inflicts capital punishment
SELECTS 1
HAS HYPONYM
SELECTS 2
{person, human being}N
Gloss: individual of the human species
HAS HYPONYM
{head}N
Gloss: the upper part of the human body ...
{guillotine}V
Event type: transition Arguments:<1, 2, 3>
Subevents: e1(1, 2, 3), e2(2)
Restrictions: e1, e2 Head: e2
Gloss: execute by beheading with a guilllotine
HAS HYPONYM
HAS PART
INCORPORATES 2
{decapitate, behead}V
Event type: transition Arguments:<1, 2, 3>
Subevents: e1(1, 2, 3), e2(3)
Restrictions: e1, e2 Head: e2
Gloss: cut the head
INCORPORATES 3, conj1:1
{with}Prep
Gloss: indicator of means
INCORPORATES 3, conj2:1
{guillotine, scaffold}N
Gloss: instrument of execution that consists of a weighted blade between two vertical poles used for beheading people
HAS TELIC
HAS HYPONYM
{cut}V
Event type: process Arguments:<1, 2, 3>
Subevents: e1(1, 2, 3) Restrictions: Head: e1
Gloss: separate in parts with an instrument
HAS PART
IS USED AS INSTRUMENT TO
HAS HYPONYM
SELECTS BY DEFAULT 3
{blade}N
Gloss: flat part of a tool or weapon with a cutting edge
{instrument}N
Gloss: object used to perform a given activity

Fig. 1. Wordnet fragment with event and argument information

tasks and applications. We furthermore show that the different modeling structures of GL and WordNet models are truly complementary and concur to a more accurate representation of the mental lexicon.

References

1. Miller, G.A.: WordNet: an Online Lexical Database. Special Issue of International Journal of Lexicography 4(3) (1990)
2. Fellbaum, C.: A Semantic Network of English: the Mother of all WordNets. In: Vossen, P. (ed.) EuroWordNet: a Multiligual Database with Lexical Semantic Networks, pp. 137–148. Kluwer Academic Publishers, Dordrecht (1998)
3. Kohl, K., Jones, D., Berwick, R., Nomura, N.: Representing Verb Alternations in WordNet. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. 153–178. The MIT Press, Cambridge (1998)
4. Leacock, C., Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. 265–284. The MIT Press, Cambridge (1998)
5. Harabagiu, S., Moldovan, D.: Knowledge Processing on an Extended WordNet. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. 353–378. The MIT Press, Cambridge (1998)

6. Agirre, E., Martinez, D.: Integrating selectional preferences in WordNet. In: Proceedings of the 1st International WordNet Conference, Mysore (2002)
7. Marrafa, P.: The Portuguese WordNet: General Architecture and Semantic Internal Relations. DELTA, Brasil (2002)
8. Vossen, P. (ed.): EuroWordNet – A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers, Dordrecht (1998)
9. Pustejovsky, J.: The Syntax of Event Structure. Cognition 41, 47–81 (1991)
10. Pustejovsky, J.: The Generative Lexicon. The MIT Press, Cambridge (1995)
11. Bentivogli, L., Pianta, E.: Extending WordNet with Syntagmatic Information. In: Sojka, P., Pala, K., Smrž, P., Fellbaum, C., Vossen, P. (eds.) Proceedings of the 2nd International WordNet Conference, pp. 47–53. Czech Republic, Brno (2004)
12. Lemnitzer, L., Wunsch, H., Gupta, P.: Enriching GermaNet with Verb-noun Relations – a Case Study of Lexical Acquisition. In: Proceedings of LREC 2008, Marrakech, Morocco, pp. 156–160 (2008)
13. Boyd-Graber, J., Fellbaum, C., Osherson, D., Schapire, R.: Adding Sense, Weighted Connections to WordNet. In: Proceedings of the 3rd International WordNet Conference. Brno, Czech Republic (2006)
14. Marrafa, P.: The Representation of Telic Complex Predicates in Wordnets: the Case of Lexical-Conceptual Structure Deficitary Verbs. Research on Computing Science 12 (2005)
15. Amaro, R., Chaves, R.P., Marrafa, P., Mendes, S.: Enriching wordnets with New Relations and with Event and Argument Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 28–40. Springer, Heidelberg (2006)
16. Marrafa, P.: Predicação Secundária e Predicados Complexos em Português. Análise e Modelização. Ph.D. dissertation. University of Lisbon, Lisbon (1993)

Lexical-Conceptual Relations as Qualia Role Encoders

Raquel Amaro, Sara Mendes, and Palmira Marrafa

CLG, Center of Linguistics of the University of Lisbon
Av. Prof. Gama Pinto, 2, 1649-003 Lisbon, Portugal
{ramaro,sara.mendes}@clul.ul.pt, palmira.marrafa@netcabo.pt
http://www.clul.ul.pt/clg

Abstract. In this paper we show how wordnets can be used for building computational lexica that support generative processes accounting for phenomena such as the creation of meaning in context. We propose the integration of qualia information in wordnets through the association of lexical-conceptual relations to qualia roles, in what is a simple and low cost procedure, as it makes use of information already encoded in wordnets. This association between lexical-conceptual relations and qualia aspects allows us to describe the qualia structure of lexical items in a consistent way, without any loss of information and with the advantage of identifying the semantic predicates that can be values of qualia roles.

Keywords: Wordnets, qualia information, lexical-conceptual relations, generative lexica.

1 Introduction

Wordnets are electronic databases developed along the general lines of the so-called Princeton WordNet, [1,2] containing nouns, verbs, adjectives, and adverbs. This database is structured as a network of relations between *synsets* (sets of synonymous word forms). Although originally developed as an experiment on the organization of the mental lexicon, WordNet has been widely used as a lexical base in NLP applications.

In this context, the need for finer-grained lexical specifications allowing computational systems to deal automatically with various complex linguistic phenomena in a general and systematic way has been pointed out by many different authors. The association of syntactic and semantic information to the WordNet model, in particular, has been object of research since its appearance (see for instance [3,4,5] or [6], among others).

One of the major achievements of research on WordNet enhancement is EuroWordNet [7], a resource reflecting research both on lexical semantics and on wordnets as lexical resources for NLP applications. Such research resulted in the definition and implementation of a wider variety of lexical-conceptual relations than the set used in the Princeton WordNet, focusing on more comprehensive lexical-conceptual relations and on cross-POS relations.

In this paper we pursue these lines of investigation, building on research developed under the scope of WordNet.PT [8] (henceforth, WN.PT) to model the semantic and syntactic properties of lexical items, while providing the relevant informational structures describing the nature of lexical meaning. Although we set off from Portuguese

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 29–36, 2010.

data, we claim that our approach is language independent and widely extendable to other languages.

The relevance of such work lies on the fact that lexical items are the basic building blocks in language, being involved in a large diversity of syntactic and semantic phenomena. Being so, in order to account for this behavior, as well as for the polymorphic properties and creative use of lexical items, the need for defining lexical entries as complex informational structures becomes apparent (see [9] and [10], among others). The Generative Lexicon [11,9] (henceforth, GL) is one of the most illustrative examples of this perspective on lexical modeling. In this framework, the Lexicon is a complex system constituting a crucial part of the grammar of natural languages. Lexical units are represented as informational structures, organized according to a finite set of rules which allow for accounting for the creation of meaning in context and for describing the relation between syntax and semantics concerning several relevant linguistic phenomena.

The different levels of representation considered in GL set the grounds for information sharing between lexical items, whereas the generative mechanisms described in this framework assure a coherent and recursive codification of the information. In this paper we focus on the first part of the problem, i.e. on contributing to an accurate description of lexical items so that such representations can serve as a suitable input to the aforementioned generative mechanisms.

Concurrently, computational relational lexica such as wordnets straightforwardly provide us with the grounds for a lexical inheritance device allowing us to encode information sharing between lexical units. Hence, this paper combines these features of wordnets and GL, this way contributing to richer lexica through a parsimonious strategy of lexical modeling.

2 GL Structures and Wordnets

Having motivated the integration of additional information in the WordNet model for increasing both its accuracy and usability as a lexical base for NLP applications, in this section we present our proposal for using information in wordnets as encoders of GL structures, more specifically for using the relations encoded in WN.PT to obtain qualia information.

2.1 Qualia Structure

Qualia structure is the level of representation in which the semantic content of a lexical item is encoded, through the properties and events that define it. There are four basic qualia roles, determining the lexical-semantic structure of lexical items:

- CONSTITUTIVE: expressing the relation between an object and its constituent parts;
- FORMAL: representing the features which distinguish an object within a larger domain;
- TELIC: stating the purpose and function of an object;
- AGENTIVE: enumerating the factors involved in the origin or ’bringing about’ of an object.

In terms of the linguistic characterization of lexical items, qualia structure establishes the set of semantic restrictions introduced by a word in context. Although these semantic restrictions can sometimes undergo semantic change – in metaphoric contexts for instance –, they explain ill/well-formation contrasts between certain structures as well as available readings. Let us briefly go through some examples to illustrate how qualia values have impact on the well-formedness of sentences.

1. a. Anne scaled the fish.
   b. *Anne scaled the chicken.
   c. *Anne plucked the fish.
   d. Anne plucked the chicken.
2. a. John started the house.
   b. = John started to build the house.
3. a. John started the book.
   b. = John started to write the book.
   c. = John started to read the book.

In (1) above, the ill-formation of (1)b and (1)c can be explained by the different values for the **constitutive role** of *chicken* and *fish*: *chicken* have *feathers* and *fish* have *scales*. Such values, in combination with the semantic restrictions introduced by the main verbs in (1) – both *to scale* and *to pluck* have shadow arguments incorporated in their semantics: *scale* and *feather*, respectively –, justify the syntactic contrasts between (1)a and (1)b, and between (1)c and (1)d.

Besides accounting for grammaticality contrasts, as illustrated in (1), qualia roles can also explain available readings. Differently from the previous example, in (2) and (3) there are no grammaticality contrasts. In this case the challenge consists in explaining how the readings in (2)b, (3)b and (3)c are derived. Qualia structure enables the association of particular properties and activities to nouns, which in turn provides the verb selecting the NP in which these nouns occur with the information for contextualising its sense. *house*, in (2), is associated to a **building event** in the **agentive role** of its qualia structure and this is how this event is made available to the reading presented in (2)b. In (3), we have two different possible readings of (3)a, in (3)b and (3)c, because *book* is associated to two events in its qualia structure: a **writing event** in the agentive role and a **reading event** in the telic role, thus making both events available.

Based on data such as these, [9] defines a set of generative mechanisms which take the information identified above to derive available readings. However, not all lexical items associated to events in their qualia structure display the linguistic behavior illustrated in (3). If we take a word like *pen*, a writing implement, we see that not all the readings in (3) are available for this lexical item, although *pen*, just like *book*, is associated to a **constructing event** in the **agentive role**, and to a **writing event** in the **telic role**. A sentence like *John is starting the pen* has a single available reading: John is starting to construct (by assembling parts or carving, for instance) a *pen*. The reading in which he is starting to write with a pen is never obtained. This clearly poses a problem to [9]'s generalization presented above. Data like this indicate that further specifications are called for, for instance in the form of qualia subtyping, or some other strategy allowing generative mechanisms to account for these contrasts. But this is a question that is clearly outside the scope of the work depicted in this paper, which we are currently addressing independently. Despite counter examples like the one discussed above, the relevance of qualia information in the characterization of the semantic contribution of lexical items is nonetheless undeniable, the impact it has on their linguistic behavior being what still calls for further analysis and specifications.

2.2 Qualia Information in Wordnets

Qualia information reflects the semantic content of the lexical item described. Being so, specifying such information in relational models of the lexicon is entirely consistent with the nature of such models. As noted in the literature (see [12,13], among others) WordNet structuring relation – *hyperonymy/hyponymy* –, for instance, refers to the **formal quale** of a lexical item, *meronymy* relations to the **constitutive quale**, *cause* relations to the **agentive quale**, and so on.

Qualia information has also been noted to be expressed indirectly in wordnets at other levels. [13], for instance, in the context of research on metaphorical uses of lexical items, puts forth a proposal consisting on automatically extracting **agentive** and **telic** information from the glosses in WordNet.

As to the EWN model, it reflects to some extent the notion of qualia information in its structure, as GL qualia structure was used for defining and building its top-ontology [14]. This top-ontology reflects a taxonomic approach to qualia structure, as it does not allow for simultaneously accounting for multiple aspects of the meaning of a specific lexical item (see [11]).

Pursuing this taxonomic approach to qualia information in EWN, [12] puts forth a proposal for distinctively modeling taxonomic hyponyms (those associated to 'canonical' *hyperonymy* relations and corresponding to the **formal quale** of the lexical item) and orthogonal hyponyms (split according to the other three qualia dimensions) in Danish wordnet. This strategy results in the distinction of *constitutive hyponyms* (*idiot* 'idiot' or *geni* 'genius'), *telic hyponyms* (*garvestof* 'tanning agent' or *flugtbil* 'getaway car') and *agentive hyponyms* (*fodgĂŚnger* 'pedestrian' or *cyklist* 'cyclist') (examples taken from [12]: 5). However, there are some stebacks to this taxonomic approach to qualia information.

In fact, although [12] accounts for phenomena such as compatible co-hyponymy (see [15,16]), it continues to lack the ability to mirror the fact that lexical items can be, and often are, simultaneously characterized by multiple qualia.

Reflecting different goals and strategies, the approaches discussed above reinforce the acknowledgement of a general need for being able to obtain more complete informational structures from lexical resources, while showing at the same time that relational models of the lexicon like wordnets convey, more or less directly, a significant amount of relevant information regarding the internal semantic content of lexical items.

2.3 Qualia Information in WordNet.PT

In previous sections we showed how wordnets are particularly well placed to satisfy such a need: their network of relations, in combination with additional information also encoded in these resources, make informationally rich lexical entries available; at the same time, the hierarchy of relations provides a natural inheritance structure, straightforwardly allowing for information sharing between lexical entries, and thus for an economical lexical encoding. We motivated the relevance of using qualia structure to characterize lexical items, and went through some proposals for integrating this level of representation in wordnets, many of which with promising results. However, most of the proposals available in the literature are partial and often oriented towards solving particular problems of a specific application, hence not addressing the question of lexical representation in a systematic and global perspective. Doing so is our goal.

Lexical-conceptual relations reflect intrinsic or prototypical properties that characterize a given concept. Having this in mind, we analyzed the relations available in WN.PT to determine whether these can be matched to express qualia properties, and which properties are these. By doing so, we established a correspondence between the lexical-conceptual relations considered in WN.PT and qualia roles, besides the general acknowledged correspondence between *meronymy* and **constitutive** properties, and *hyponymy* and **formal** properties.

In the table presented below we show how the relations in wordnets developed in the EWN model can be used to systematically express all qualia roles. Achieving this requires only the definition and codification of two new relations in wordnets. As aforementioned the *is hyponym of* and *has as part* relations express the generic cases of **formal** and **constitutive** properties, respectively. Being so, it is only necessary to define relations to express the generic case of **agentive** and **telic** properties. Thus, we introduce the relations *results/originates from* and *has as function/goal* to fill in this gap. Their respective counterparts – *results in/originates* and *is function/goal of* –, however, reflect **formal** properties.

4. RESULTS/ORIGINATES FROM/RESULTS IN/ORIGINATES relation
   {synset}1 RESULTS/ORIGINATES FROM {synset}2 and {synset}2 RESULTS IN/ ORIGINATES {synset}1

5. HAS AS FUNCTION/GOAL/IS FUNCTION/GOAL OF relation
   {synset}1 HAS AS FUNCTION/GOAL {synset}2 and {synset}2 IS FUNCTION/ GOAL OF {synset}1

The Table 1 on pages 34–35 summarizes our analysis, grouping WN.PT relations, in the second column, that correspond to values of the same qualia role. Given the quality of the information stated through each specific relation, the two directions of a same relation can refer to different qualia aspects, as can be observed above. Also these relations[1] constitute values of qualia roles if, and only if, they are canonical relations defining the meaning of a lexical item, i.e. if not marked by the reversed feature (indicator of inverse relation by default).

As made apparent in the table presented above, we claim that the integration of qualia roles in wordnets is a straightforward and low cost procedure since lexical-conceptual relations in wordnets already encode intrinsic or prototypical properties that characterize the concept lexicalized by each synset. This way, associating the relevant relations to a given qualia aspect allows us to encode the qualia information in a coherent and consistent way, without any loss of information. This approach also has the advantage of determining which semantic predicates can be values of qualia roles.

[1] As some of the labels used in WN.PT were changed to be more transparent for non-specialist users, we present their corresponding relations in EWN. Regarding WN.PT-specific relations, their relevance has been argued for in [17,18].

Table 1. WN.PT relations as qualia roles encoders

WN.PT	WN/EWN	EXAMPLE
Formal role		
is hyponym (subtype) of	*has hyperonym*	{animal}N→ {living being}N
has as a characteristic	–	{carnivorous}Adj→ {shark}N
is related to	–	{marine}Adj→ {sea}N
has manner	*has manner*	{sway}V→ {smoothly}Adv
is part of	*has holonym*	{brick}N→ {wall}N
is individuated part of	*has holo part*	{motor}N→ {machine}N
is portion of	*has holo portion*	{drop}N→ {liquid}N
is member of	*has holo member*	{member}N→ {club}N
is substance/material of	*has holo made of*	{fiber}N→ {tissue}N
is sublocation of	*has holo location*	{palm}N→ {hand}N
is subevent of	*is subevent of*	{inhale}V→ {breathe}V
is telic subevent of	–	{suspicious}Adj→ {suspect}V
characterizes with regard to	*is value of*	{big}Adj→ {size}N
sets attribute value to	–	{tall}Adj→ {plus}Adv
co relates with	*co_role*	{roof tile}N→ {roof lattice}N
causes	*causes*	{kill}V→ {die}V
has as result	*involved result*	{build}V→ {building}N
has as location	*involved location*	{swim}V→ {liquid}N
has as source location	*involved source direction*	{unbox}V→ {box}N
has as goal location	*involved target direction*	{emigrate}V→ {foreign country}N
results in/originates	–	{architecture}N→ {blueprint}N
is function/goal of	–	{learn}V→ {pupil}N
Telic role		
has as function/goal	–	{pupil}N→ {learn}V
has telic subevent	–	{sadden}V→ {sad}Adj
is the instrument used for	*role instrument*	{pen}N→ {write}V
is the location for	*role location*	{pool}N→ {swim}V
is the source location of	*role source direction*	{start line}N→ {race}N
is the goal location of	*role target direction*	{prison}N→ {incarcerate}V
is transformed in	*co_patient result*	{tadpole}N→ {frog}N
is used to obtain	*co_instrument result*	{coffee grinder}N→ {coffee powder}N
acts to obtain	*co_agent result*	{lumberjack}N→ {wood}N
relates as agent with the object	*co_agent patient*	{lumberjack}N→ {tree}N
relates as object with the agent	*co_patient agent*	{brick}N→ {mason}N
uses as instrument	*co_agent instrument*	{mason}N→ {trowel}N
is used as instrument by	*co_instrument agent*	{trowel}N→ {mason}N
relates as object with the instrument	*co_patient instrument*	{mortar}N→ {trowel}N
relates as instrument with the object	*co_instrument patient*	{trowel}N→ {mortar}N

Table 1. (*continued*)

WN.PT	WN/EWN	EXAMPLE
Agentive role		
results/originates from	–	{blueprint}N→ {architecture}N
is caused by	*is caused by*	{find}V→ {search}V
is the result of	*role result*	{roastbeef}N→ {roast}V
results from the transformation of	*co_result patient*	{frog}N→ {tadpole}N
results from the use of	*co_result instrument*	{coffee}N→ {coffee pot}N
results from the action of	*co_result agent*	{bread}N→ {baker}N
Constitutive role		
has as part	*has meronym*	{wall}N→ {brick}N
has as individuated part	*has mero part*	{car}N→ {wheel}N
has as portion	*has mero portion*	{liquid}N→ {drop}N
has as member	*has mero member*	{club}N→ {member}N
has as substance/material	*has mero madeof*	{coffee}N→ {caffeine}N
has as sublocation	*has mero location*	{hand}N→ {palm}N
has subevent	*has subevent*	{breathe}V→ {inhale}V
has telic subevent	–	{sadden}V→ {sad}Adj

3 Final Remarks

In this paper we show how the specification of fine-grained lexical descriptions in computational relational lexica provides the grounds for accounting for several lexical semantic phenomena. Our strategy contributes to enhancing wordnets usability as computational lexica supporting generative processes to account for phenomena such as the creation of meaning in context.

In our proposal, the association of qualia roles to synsets in wordnets is achieved very straightforwardly. This procedure, which hardly adds any costs to the effort in wordnets development, as it only involves the definition of two additional relations, allows for determining the semantic predicates that can be values of qualia roles in a coherent and consistent way. This strategy provides us with the mechanisms for describing and encoding crucial semantic properties of lexical items, providing the relevant information at the lexical level.

References

1. Miller, G.A.: WordNet: an Online Lexical Database. Special Issue of International Journal of Lexicography 4(3) (1990)
2. Fellbaum, C.: A Semantic Network of English: the Mother of all WordNets. In: Vossen, P. (ed.) EuroWordNet: a Multiligual Database with Lexical Semantic Networks, pp. 137–148. Kluwer Academic Publishers, Dordrecht (1998)
3. Kohl, K., Jones, D., Berwick, R., Nomura, N.: Representing Verb Alternations in WordNet. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. 153–178. The MIT Press, Cambridge (1998)

4. Leacock, C., Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. 265–284. The MIT Press, Cambridge (1998)
5. Harabagiu, S., Moldovan, D.: Knowledge Processing on an Extended WordNet. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. 353–378. The MIT Press, Cambridge (1998)
6. Agirre, E., Martinez, D.: Integrating Selectional Preferences in WordNet. In: Proceedings of the First International WordNet Conference, Mysore (2002)
7. Vossen, P. (ed.): EuroWordNet – A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers, Dordrecht (1998)
8. Marrafa, P.: The Portuguese WordNet: General Architecture and Semantic Internal Relations. DELTA, Brasil (2002)
9. Pustejovsky, J.: The Generative Lexicon. The MIT Press, Cambridge (1995)
10. Jackendoff, R.: Semantic Structures. The MIT Press, Cambridge (1990)
11. Pustejovsky, J.: The Syntax of Event Structure. Cognition 41, 47–81 (1991)
12. Pederson, B., Sørensen, N.: Towards Sounder Taxonomies in WordNets. In: Ontolex 2006, Genova, Italy, pp. 9–16 (2006)
13. Veale, T.: Qualia Extraction from WordNet. In: Proceedings of the 3rd International Workshop on Creative Systems, The 2003 International Joint Conference on Artificial Intelligence, Acapulco, Mexico (2003)
14. Vossen, P. (ed.): EuroWordNet – A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers, Dordrecht (1998)
15. Mendes, S., Chaves, R.P.: Enriching WordNet with Qualia Information. In: Proceedings of the Workshop on WordNets and Other Lexical Resources at NAACL 2001 Conference, Pittsburgh, pp. 108–112 (2001)
16. Amaro, R., Chaves, R.P., Marrafa, P., Mendes, S.: Enriching Wordnets with new Relations and with Event and Argument Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 28–40. Springer, Heidelberg (2006)
17. Marrafa, P.: The Representation of Telic Complex Predicates in Wordnets: the Case of Lexical-Conceptual Structure Deficitary Verbs. Research on Computing Science 12 (2005)
18. Mendes, S.: Syntax and Semantics of Adjectives in Portuguese: Analysis and Modeling. Ph.D. dissertation, University of Lisbon, Lisbon (2009)

Towards Disambiguation of Word Sketches

Vít Baisa

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
xbaisa@fi.muni.cz

Abstract. A word sketch is a source of valuable information both for linguists and lexicographers but it consists of lemmas which are not disambiguated. In this paper we describe a method which can partially disambiguate these lemmas and increase a quality of information contained in word sketches. For the disambiguation we exploit intersections of English and Czech word sketches using an English-Czech dictionary.

Keywords: Word sketch, English-Czech dictionary, Sketch Engine, partial disambiguation.

1 Introduction

A word sketch of a word is a corpus-based summary of the word's grammatical and collocational behaviour. The word sketch consists of grammatical relations that the word participates in.

Word sketches are derived from a text corpus on a basis of word sketch grammars. Each relation is defined by appropriate word sketch grammar rule. These rules are CQL queries and represent patterns for matching collocations. Table 1 shows abridged word sketch for word *key*.

Table 1. Abridged word sketch for lemma *key*

a_modifier	object_of	n_modifier	modifies	modifier
alt	press	arrow	stage	manually
allen	hold	encryption	role	absolutely
cryptographic	steal	cursor	element	together
programmable	hit	shortcut	issue	alike
primary	turn	ignition	stakeholder	chiefly
private	assign	description	theme	largely
minor	bend	backspace	area	forward
golden	obtain	activation	aspect	increasingly
lost	define	hash	principle	however
128-bit	insert	ctrl	question	especially
F11	enter	F	figure	perhaps

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 37–42, 2010.

A kind of each relation should be clear. The word *key* `modifies` word *role* in collocation “key role”, word *golden* is an adjective modifier (`a_modifier`) of word *key* in collocation “golden key”, word *cursor* is a noun modifier (`n_modifier`) and so on.

Obviously, words from these relations collocate with distinct senses of the lemma *key*. Since *key* is polysemous, there are for example two collocations “minor key” and “programmable key” (see relation `a_modifier`) in which lemma *key* refers to two distinct senses.

Our goal is a classification of words from word sketches according to their senses.

2 Principle

The classification consists of two steps: creating classes and assigning words from a relation to these classes. In our case a class is an approximation of a word sense.

We create these classes using an English-Czech dictionary. An English word connected with one of its Czech equivalents serves as a class. Despite some of these equivalents may be synonyms each other they rather represent distinct senses of the English word.

The second step is based on presumption that an English word has similar collocates in an English corpus as its Czech equivalent word has in a Czech corpus. If the English word is polysemous and if it has Czech equivalents with distinct senses, these Czech equivalents may have distinct collocates. Comparison of Czech collocates with collocates of the English word then may serve for the assigning to classes and therefore for the partial disambiguation.

The principle of our method applied on lemma *key* and relation `a_modifier` is depicted on Figure 1.

The smaller circle contains words from relation `a_modifier` – English adjective collocates (lemmas) of *key*. Let us call them *in-words*. Three grey pieces in the annulus represent three classes: pairs *key–klávesa*: (a key on a keyboard), *key–klíč* (a wrench, a clue, a key for locking, coding or encrypting) and *key–tónina* (tonality).

It must be emphasized that classes are approximations of senses because a dictionary sometimes does not contain equivalents for all senses of an English word and moreover both words in a pair may be polysemous (see the pair *key–klíč*).

Outside the bigger circle there are Czech adjective collocates (words from relation `a_modifier`) of three appropriate Czech equivalents. Let us call them *out-words*.

An *in-word* is connected with an *out-word* by a link only if they are equivalent, i.e. if they are translatable by used English-Czech dictionary. Links connecting *in-words* and *out-words* within at most one grey piece are *unique* and the other links (see dashed links on the figure) are *non-unique*.

If we encounter collocation “lost key” we can not say whether it refers to a lost key for opening a lock or to a lost key on a keyboard. In Czech language there are collocations “ztracený klíč” and “ztracená klávesa” (Czech translations of two English senses respectively) and that is why these collocations generate *non-unique* links: *ztracený* and *lost* are connected by links within both pairs *key–klíč* and *key–klávesa*.

On the contrary, in collocation “cryptographic key” and “minor key” we know that the former refers to sense expressed by Czech equivalent *klíč* and the latter to sense

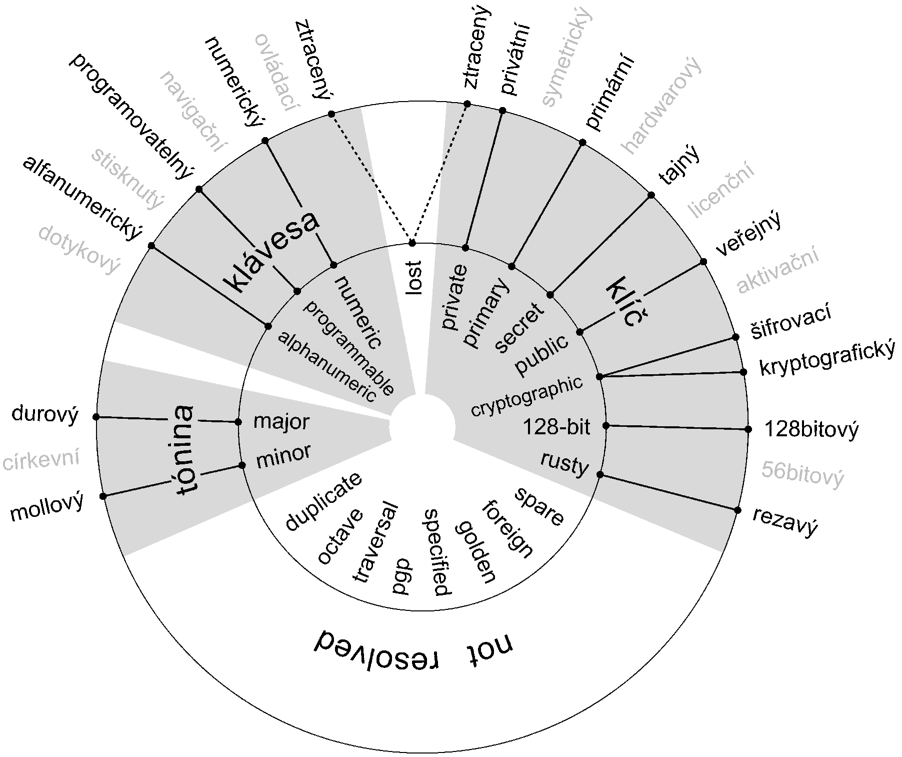


Fig. 1. Classification of relation `a_modifier` for lemma *key*

expressed by Czech word *tónina*. In used Czech corpus there are not collocations *mollový klíč* (a minor key for locking) or *kryptografická tónina* (a cryptographic tonality) hence appropriate links are *unique*.

Unique links play key role in the classification. Three grey pieces overlap the smaller circle (corresponding to appropriate *unique* links) and dissect it. The dissection of the circle is the classification of the relation. *In-words* not connected to any of *out-words* are unresolved.

3 Algorithm

We used English corpus ukWaC [1] and Czech corpus CZES for deriving word sketches by the Sketch Engine [2]. GNU/FDL English-Czech dictionary [3] was used for translations and creating classes.

Since there has been the only common relation (`a_modifier`) between English and Czech word sketch grammars used for ukWaC and CZES and since this relation captures collocations of nouns with adjectives, we restricted word sketches only to those for English nouns.

We used only one-word English nouns since the Sketch Engine can not handle word sketches of multi-word expressions. Proper nouns were also filtered. A process of computing links for an English noun consists of these steps:

1. the noun is processed only if it is covered by the dictionary,
2. a word sketch for the noun is retrieved by the Sketch Engine, namely first (topmost) 30 words from the relation `a_modifier`, let us call them *an English set*,
3. the noun is translated into Czech (possible several Czech equivalents), only Czech nouns are taken into account,
4. a word sketch for each such Czech noun is made (topmost 30 words from the relation `a_modifier` – *Czech sets*),
5. items of the English set are translated into Czech,[1]
6. intersections of the translated English set with each Czech set are computed: items present in exactly one of these intersections are *unique links*.

4 Results

Table 2 shows some statistics of the English-Czech dictionary. These numbers hold for English nouns (restricted in the same way as before: only one-word expressions without proper nouns).

Table 2. Basic statistics of used dictionary

number of words	28,209
number of polysemous words	9,149
average number of Czech equivalents per word	1.55
average number of Cz. eq. (only polysemous words)	2.69
maximum number of Czech equivalents (*line*)	22

It is worth comparing these numbers with the WordNet statistics [4]. Average number of senses per a noun in the WordNet equals 1.24 (2.79 excluding monosemous nouns). There are 15,935 polysemous nouns in the WordNet. Moon [5] mentions about 10,000 English polysemous words.

Results of our method are listed in Table 3. We have processed 15,025 words with 1.9 Czech equivalents on average. 6,328 of them were polysemous with 4.52 Czech equivalents per word on average. The process linked on average 18.58 words within relation `a_modifier` from a word sketch of a polysemous English word with their Czech equivalents and 8.09 of these links were *unique* on average. Appropriate numbers for all words (including monosemous words) are slightly worse: 6.28 *unique* links out of all 10.69 links, both on average.

Three nouns are worth mentioning: *line*, *part* and *vein*. The first one has the highest number of Czech equivalents in the dictionary (22 Czech equivalents, 30 noun synsets

[1] Situations in which an item has more than one Czech equivalent are almost harmless. They just extend the English set.

Table 3. Overall results

number of retrieved words	15,025
number of retrieved polysemous words	6,328
average number of Czech equivalents per word	1.9
average number of Czech equivalents per polysemous word	4.52
average number of links per word	10.69
average number of *unique* links per word	6.28
average number of links per polysemous word	18.58
average number of *unique* links per polysemous word	**8.09**
maximum number of links (*part*)	133
maximum number of unique links (*vein*)	25
number of polysemous nouns without unique links	66

in the WordNet [6]). The second one (*part*) has the highest number (133) of links. It is caused by both its high polysemy (14 Czech equivalents, 12 noun synsets in the WordNet) and potential to collocate with general adjectives – just 14 of these 133 links are *unique*. The last one (*vein*) is the most suitable for the partial disambiguating: our method is able to classify 83.33 % of relation `a_modifier`.

Only 66 of English polysemous nouns have no *unique* links: *excoriation, indaba, neb, grump,*... Their Czech equivalents are rare in the Czech corpus and rather mutually synonymic. Synonymic words have very similar collocates which are shared among classes and that is why they can not generate *unique* links. Nevertheless it is 1 % of all examined polysemous nouns.

5 Conclusions and Future Work

We obtained 8.09 *unique* links per word on average out of all 30 nouns from the relation. That means we are able to partially disambiguate about 1/4 of the relation in the corpus properly. It looks promising since we have used just one relation out of 25 relations defined by Czech and English word sketch grammars.

It is question (and matter of our future work) whether involving other languages would produce more accurate disambiguation. A similar question concerns using parallel corpora.

We rely on the only relation (`a_modifier`). Hence the most important and immediate goal is to develop new Czech and English word sketch grammars more suitable for our purpose: to capture the same and the most similar grammar phenomena occurring simultaneously in both languages. With such English-Czech word sketch grammar in our hands we believe in further improvement of already favourable results.

Acknowledgement

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536.

References

1. Ferraresi, A., Zanchetta, E., Baroni, M., Bernardini, S.: Introducing and Evaluating ukWaC, a Very Large Web-Derived Corpus of English. In: Proceedings of the 4th Web as Corpus Workshop (WAC-4), Marrakech, Morocco (2008)
2. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch Engine. In: Proceedings of the Eleventh EURALEX International Congress, pp. 105–116. Universite de Bretagne-Sud, Lorient (2004)
3. Svoboda, M.: GNU/FDL English-Czech Dictionary (2001), http://slovnik.zcu.cz
4. WordNet Statistics, http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html
5. Moon, R.: Lexicography and Disambiguation: The Size of the Problem. In: Computers and the Humanities, vol. 34, pp. 99–102. Springer, Netherlands (2000)
6. WordNet, http://wordnetweb.princeton.edu/perl/webwn

Towards an N-Version Dependency Parser

Miguel Ballesteros[1], Jesús Herrera[1], Virginia Francisco[1], and Pablo Gervás[2]

[1] Departamento de Ingeniería del Software e Inteligencia Artificial
[2] Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
C/ Profesor José García Santesmases, s/n E–28040 Madrid, Spain
{miballes,jesus.herrera,virginia}@fdi.ucm.es, pgervas@sip.ucm.es

Abstract. Maltparser is a contemporary dependency parsing machine learning-based system that shows great accuracy. However 90% for Labelled Attachment Score (LAS) seems to be a *de facto* limit for such kinds of parsers. Since generally such systems can not be modified, previous works have been developed to study what can be done with the training corpora in order to improve parsing accuracy. High level techniques, such as controlling sentences' length or corpora's size, seem useless for these purposes. But low level techniques, based on an in-depth study of the errors produced by the parser at the word level, seem promising. Prospective low level studies suggested the development of n-version parsers. Each one of these n versions should be able to tackle a specific kind of dependency parsing at the word level and the combined action of all them should reach more accurate parsings. In this paper we present an extensive study on the usefulness and the expected limits for n-version parser to improve parsing accuracy. This work has been developed specifically for Spanish using Maltparser.

1 Introduction

In the 10th edition of the Conference of Computational Natural Language Learning (CoNLL) a first shared task on Multilingual Dependency parsing was accomplished [1]. Thirteen different languages including Spanish were involved and parsing performance was studied. In this Shared Task, participants implemented a parsing system that could be trained for all these languages. Maltparser 0.4 is the publicly available software that is contemporary of the system presented by Nivre's group to the CoNLL-X Shared Task, in which Spanish was proposed for parsing and Nivre's group achieved great results.

Dependency parsing machine learning-based systems show exceptional accuracy. However 90% for Labelled Attachment Score (LAS) seems to be a *de facto* limit for such kinds of parsers. Since generally such systems can not be modified, we developed some works to study what can be done with the training corpora in order to improve parsing accuracy. High level techniques, such as controlling sentences' length or corpora's size, seem useless for these purposes. However they appeared useful for the design of systematic processes for building training corpora [2]. Low level techniques, based on an in-depth study of the errors produced by the parser at the word level, seem promising. Prospective low level studies suggested the development of n-version parsers. Each one of these n versions should be able to tackle a specific kind of dependency parsing at the word level and the combined action of all them should

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 43–50, 2010.

reach more accurate parsings. Since n-version parsers could be a valid tool for improving parsing accuracy, we present in this paper an in-depth study on their usefulness and expected limits, as a continuation of our previous work described in [3].

The paper is organized as follows: Section 2 describes the CoNLL-X Shared Task focusing on Spanish participation. In Section 3 we describe the n-version parsing model developed. In Section 4 we analyze the values obtained both for local accuracy and overall accuracy. Finally, Section 5 shows the conclusions of the presented work and suggests some future work.

2 The CoNLL-X Shared Task

The goal of the CoNLL-X Shared Task [1] was to label dependency structures by means of a fully automatic dependency parser. This task provided a benchmark for evaluating parsers accross 13 languages, one being Spanish. Systems were scored by computing their Labelled Attachment Score (LAS), i.e. the percentage of "scoring" tokens for which the system had predicted the correct head and dependency label [4], their Unlabelled Attachment Score (UAS), i.e. the percentage of "scoring" tokens for which the system had predicted the correct head [5] and their Label Accuracy (LA), i.e. the percentage of "scoring" tokens for which the system had predicted the correct dependency label [6].

The results for Spanish across the 19 participants ranged from 47% to 82.3% LAS, with an average of 73.5%. The treebank used was AnCora [7,8]. The two participant groups with the highest total score for Spanish were [9] and [10] with 82.3% and 81.3% LAS, respectively. We are especially interested in Nivre's group research because we used their system (Maltparser 0.4) for the experiments presented in this paper and in our previous ones on improving parsing accuracy [2,3]. The evaluation shows that the approximation given by Nivre gives competitive parsing accuracy for the languages studied. More specifically Spanish parsing scored 81.3% LAS; it was only 1 point under the best one [9], which did not use the Nivre algorithm but an Eisner's bottom-up span algorithm.

In our work, the first step was to replicate the participation of Nivre's group in the CoNLL-X Shared Task for Spanish [3]. We obtained the same results as Nivre's group, i.e., LAS = 81.30%, UAS = 84.67% and LA = 90.06%. These results served as a baseline for this work to determine ways to improve them.

3 The Development of N Specific Parsers

Considering the baseline experiment described in Section 2, despite a high overall parsing accuracy only 358 wordforms of the test corpus obtain a 100% LAS, UAS and LA in all parsed sentences, i.e., only 6.3% of the wordforms. If considering sentences, only 38 sentences of the test corpus (18.4% of them) were parsed without errors. An end user should usually expect a high local parsing accuracy (at the sentence level) rather than a high overall parsing accuracy. But nowadays a remarkable percentage of sentences in Spanish shows almost one error when parsed by Maltparser.

As described in [3], when analizing the results after parsing the test corpus, we found that there is a small set of words that show an incorrect attachment, labelling or both. These words are the prepositions “a” (*to*), “de” (*of*), “ en” (*in*), “con” (*with*), “por” (*for*), the conjunction *and*, which has two wordings: “y” or “e”, and the nexus “que” (*that*). For instance there are 20 sentences (340 wordforms), in the test corpus presented in Section 2, with only one error after parsing. That is 9.7% of the corpus’ sentences and 5.98% of its wordforms. We found that in 10 of these 20 sentences the only failure is caused by one of the words listed above.

Our hypothesis is that by enhancing local accuracy, not only overall accuracy should be enhanced, but end user satisfaction should be increased. We carried out a set of experiments to confirm or reject this hypothesis. The basic idea was to do an in-depth study of each one of the words listed above. This study, as described in [3], identified the set of different cases in which each word could be attached and labelled and train a specific parser for each case found. By doing so, we analyzed the conjunction and the preposition “a” in order to determine the feasibility of the technique. We found four different cases in which the conjunction could be attached and labelled, and six cases for the preposition “a”. So we trained 10 different specific parsers for covering the set of cases given for the conjunction and the preposition “a”. After this, the test set was parsed by combining the action of the parser described in Section 2 and the other 10 specific parsers. This way, when parsing a conjunction or a preposition “a”, the output of the general parser was ignored and was substituted by the output given by the specific parser for the given case. So the attachment and the label given for this word by the general parser were substituted by the attachment and the label given by the specific one. By doing so, overal LAS was increased by 0.87%, UAS by 0.84% and LA by 0.26%. These results encouraged us to continue with the experiment by training specific parsers for the rest of the words listed previously. The results obtained for all these words are shown in Table 1. They are usually better when using a specific parser than when using the general parser described in Section 2. But sometimes the specific parsers reach the same accuracy than the general parser, so it does not make sense to use the specific parser in such cases. For instance, when parsing the word *de* when attached to an adjective or an adverb, both the general parser and the specific parser show 100% LAS. Only when the word *y* (or *e*) acts as a nexus in coordinated copulative sentences could we not find a specific parser better than the general parser (the general parser reaches 81.3% $\text{LAS}_{y/e}$ and the specific parser reaches 75% $\text{LAS}_{y/e}$). In 21 of the 28 identified cases it was found better to use the specific parsers. Further research may produce better results for the specific parsers that do not reach 100% LAS yet.

In some cases the given improvement seems quite impressive. For instance, when parsing the word *de* when attached to a verb, the general parser shows 0% LAS and the specific parsers show 100% LAS. It is due to the little amount of samples present in the test corpus. For instance, if the test set contains only one sample for a specific case and this sample is correctly parsed then LAS = 100%. But it does not mean that the parser will parse every given sample of this case with 100% LAS. For the given example the test corpus contained only 4 samples. All these samples where wrongly parsed by the general parser but perfectly parsed by the two involved specific parsers. So LAS was enhanced from 0% to 100%, but this is for the given test corpus. If the test

Table 1. Attachment and labelling of all the studied words in AnCora. Found cases and specific LAS for each word and case, before and after the application of our method. The left arrow ($^{\leftarrow}$) after a part of speech indicates that this part of speech is before the considered word in the sentence. The right arrow ($^{\rightarrow}$) indicates that the part of speech is after the word.

		Case					
Word		#1	#2	#3	#4	#5	#6
y/e	**Label**	–	–	–	–		
	Attached to a	verb$^{\leftarrow}$	proper noun$^{\leftarrow}$	common noun$^{\leftarrow}$	adjective$^{\leftarrow}$		
	LAS$_{y/e}$ before	***81.3%***	80%	66.7%	80%		
	LAS$_{y/e}$ after	75%	***100%***	***80%***	***100%***		
a	**Label**	CD	CI	CC	CREG	–	–
	Attached to a	verb$^{\leftarrow}$					noun$^{\leftarrow}$
	LAS$_{a}$ before	62.5%	42.9%	60%	25%	***0%***	50%
	LAS$_{a}$ after	***87.5%***	***100%***	***100%***	***75%***	***0%***	***100%***
de	**Label**	CC	CREG	–	–		
	Attached to a	verb$^{\leftarrow}$		adverb$^{\leftarrow}$ adjective$^{\leftarrow}$	noun$^{\leftarrow}$		
	LAS$_{de}$ before	0%	0%	***100%***	83.3%		
	LAS$_{de}$ after	***100%***	***100%***	***100%***	***96.7%***		
que	**Label**	SUJ	–	SUJ			
	Attached to a	verb$^{\rightarrow}$		verb$^{\leftarrow}$			
	LAS$_{que}$ before	88.5%	86.4%	0%			
	LAS$_{que}$ after	***92.3%***	***95.5%***	***100%***			
en	**Label**	CC	CC	CREG	–		
	Attached to a	verb$^{\rightarrow}$	verb$^{\leftarrow}$		noun$^{\leftarrow}$		
	LAS$_{en}$ before	***83.3%***	92.6%	50%	62.5%		
	LAS$_{en}$ after	***83.3%***	***100%***	***100%***	***87.5%***		
con	**Label**	CC	CREG	–	–		
	Attached to a	verb$^{\leftarrow}$			noun$^{\leftarrow}$		
	LAS$_{con}$ before	60%	40%	***100%***	66.7%		
	LAS$_{con}$ after	***80%***	***100%***	***100%***	***83.3%***		
por	**Label**	–	CAG	CAG			
	Attached to a	noun$^{\leftarrow}$	comma$^{\leftarrow}$	adjective$^{\leftarrow}$			
	LAS$_{por}$ before	***100%***	***100%***	80%			
	LAS$_{por}$ after	***100%***	***100%***	***100%***			

corpus had contained more samples perhaps the specific parsers could not have reached 100% LAS. Usually the local improvement reached by the specific parsers is very high, but as said before it must be considered cautiously because of the limited amount of samples in our test corpus, that usually are between 2 and 10 for each case, being 30 the maximum. Nevertheless, as said in [2], parsing accuracy is reasonably homogeneus and similar accuracies should be expected even when increasing the number of samples in the test set.

In addition, we found that the word *de* attached to a verb with the label "–" is a given case in the training corpus that is not given in the test corpus. Of course, for this situation no error is given by the general parser, but how can we know if the parser can tackle such a situation if it is not present in the test corpus? This is because, if we want to obtain a high performing parser we must carefully build the train and test corpora.

4 Overall Accuracy, Local Accuracy and Their Limits

As seen in [3] and in Section 3, as a result of the use of specific parsers local accuracy can be improved and this redounds to the improvement of the overall accuracy. Dependency parsers can be useful for human end users, that presumably would use such parsers to analyze little pieces of text. So end users would feed dependency parsers with isolated sentences. In this case, even a single error in the parsing of one sentence is not acceptable. This is because the developers of dependency parsers should care for a high local accuracy. After parsing the test corpus with our n-version parser we got that 42 (20.3%) of the parsed sentences show no parsing errors, while 38 (18.4%) of them where perfectly parsed with the general parser. This improvement of the local accuracy, as shown in [3], has as a consequence not only a better experience for human end users but an improvement of the overall accuracy. When parsing the test corpus by combining the action of the general parser and our proposed specific parsers, we obtained the following results for overall accuracy: LAS = 82.68%, UAS = 85.73% and LA = 90.84%. It means an improvement of 1.38% LAS, 1.06% UAS and 0.78% LA in overall accuracy with respect to the results of the general parser alone.

N-version parsers are a way to improve parsing accuracy by systematically avoiding the errors given by a general parser. Nevertheless our experiments show a slight improvement. This improvement is bigger when eliminating the errors caused by a frequent word, as shown in Figure 1, 2 and 3. In each figure, each set of bars shows the increments of LAS, UAS and LA when adding the action of specific parsers for each word considered. The first word for which we added the action of its specific parsers was the conjunction (*y* o *e*). This is because the conjunction was most frequently parsed wrong by the general parser. Following this idea, we cumulatively added the action of specific parsers for each one of the considered words, firstly those that caused more parsing errors when using the general parser. In the end, when adding the action of specific parsers for the word *por*, we got the action in synergy of all the specific parsers listed in Table 1 and the general parser. We can observe in Figure 1, 2 and 3 that LAS, UAS and LA increased notably when adding the action of specific parsers for the conjunction and for the preposition *a*. In fact LA did not increase when the action of a specific parser for the conjunction was added, but this is because the general parser did not fail when attaching the conjuntion. Thus, the specific parsers could not improve this perfect attachment. In any case, in general terms the more infrequent the word that causes parsing errors the less the contribution of its specific parsers to the overall action. So the effort for building specific parsers may not be worth the obtained improvement. It is of interest to note that the conjunction causes 56 parsing errors with the general parser, *a* causes 48 errors, *de* 44 errors, *que* 42 errors, *en* 37 errors, *con* 17 errors and *por* 16 errors. Also, the increments obtained are not regular and this is because of the number of samples of each considered case present in the test corpus and the accuracy of their specific parsers.

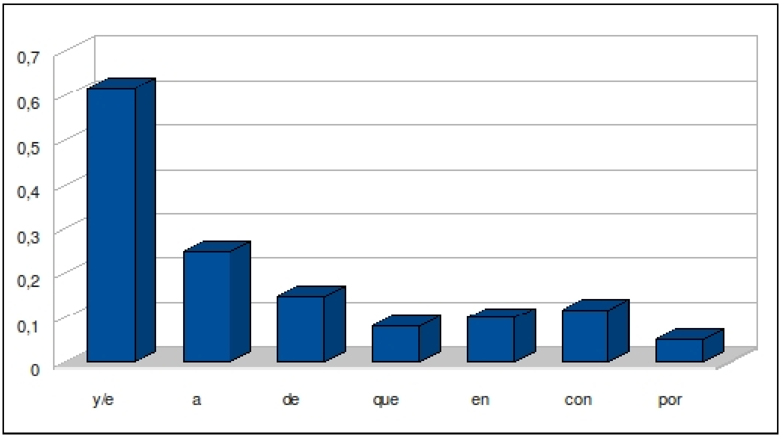


Fig. 1. Increments of overall LAS due to the action of specific parsers that avoid the more frequent errors, given by certain words

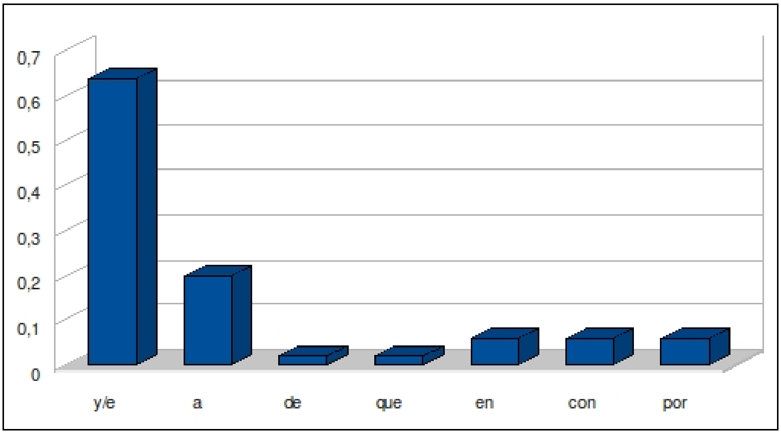


Fig. 2. Increments of overall UAS due to the action of specific parsers that avoid the more frequent errors, given by certain words

Although a big percentage of the more frequent errors given are eliminated with the n-version parser, a significate number of errors remains. After our efforts, a 17,32% improvement in LAS is still required to reach a perfect parsing. Since specific parsers have been developed only for a small set of words, some other words remain without a specific parsing solution and continue causing errors. This means that to reach a perfect parsing an additinal significant effort is needed. A lot of errors could be avoided by implementing more complex n-version parsers, covering a large number of "difficult" words than the ones presented here. But some other errors could be inherent to the implementation of Maltparser and can not be avoided. Also, as suggested in [2] and in Section 3, some other errors could be avoided by carefully building the training corpora.

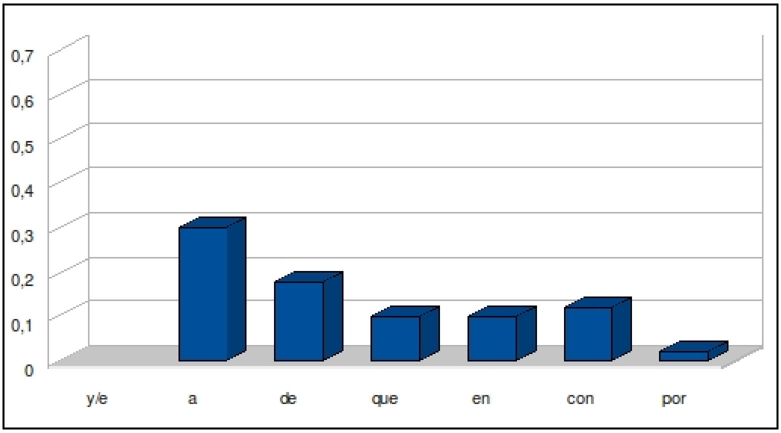


Fig. 3. Increments of overall LA due to the action of specific parsers that avoid the more frequent errors, given by certain words

5 Conclusions and Future Work

In the present paper we show that n-version parsers are useful for improving dependency parsing accuracy in the case of machine learning-based systems.

We developed a n-version parser that improved the performance of a general parser alone. To do this we identified the seven words that were most frequently parsed incorrectly by the general parser. After this, we found the set of cases in which these words were given in the corpus and we trained Maltparser 0.4 to obtain a specific parser for each case. The improvements of this n-version parser are 1.38% LAS, 1.06% UAS and 0.78% LA better than the results of the general parser. Although it means a slight improvement was acquired, n-version parsers appear to be a useful method when developing high performing dependency parsers. But n-version parsers are not the definitive solution – they must be used in synergy with a systematic developement of training and test corpora and the improvement of the implementation and settings of machine learning-based dependency parsing generators. These results are statistically significant because we only focused in a small set of words. Also, it is important to notice that by improving the parsing of those words, more well-formed dependency trees are given. This is specially useful when a word, such as prepositions, that is the head of a subtree is correctly attached. By doing so all the subtree will be correctly attached.

Future work may be a more in-depth research on n-version parsers and the implementation of programs that must accurately send each word to the more appropriated specific parser.

Furthermore, this work which has focused on Spanish language using Maltparser 0.4 could similarly be applied for parsing other languages.

Acknowledgments

This work has been partially funded by *Banco Santander Central Hispano* and *Universidad Complutense de Madrid* under the *Creación y Consolidación de Grupos de Investigación* program, Ref. 921332–953.

References

1. Buchholz, S., Marsi, E.: CoNLL-X shared task on Multilingual Dependency Parsing. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X), pp. 149–164 (2006)
2. Ballesteros, M., Herrera, J., Francisco, V., Gervás, P.: Improving Parsing Accuracy for Spanish using Maltparser. Journal of the Spanish Society for NLP (SEPLN) 44 (2010)
3. Ballesteros, M., Herrera, J., Francisco, V., Gervás, P.: A Feasibility Study on Low Level Techniques for Improving Parsing Accuracy for Spanish Using Maltparser. In: Konstantopoulos, S., Perantonis, S. (eds.) SETN 2010. LNCS (LNAI), vol. 6040, pp. 39–48. Springer, Heidelberg (2010)
4. Nivre, J., Hall, J., Nilsson, J.: Memory-based Dependency Parsing. In: Proceedings of CoNLL 2004, Boston, MA, USA, pp. 49–56 (2004)
5. Eisner, J.: Three New Probabilistic Models for Dependency Parsing: An Exploration. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING 1996), Copenhagen, pp. 340–345 (1996)
6. Yamada, H., Matsumoto, Y.: Statistical Dependency Analysis with Support Vector Machines. In: Proceedings of International Workshop of Parsing Technologies (IWPT 2003), pp. 195–206 (2003)
7. Palomar, M., Civit, M., Díaz, A., Moreno, L., Bisbal, E., Aranzabe, M., Ageno, A., Martí, M., Navarro, B.: 3LB: Construcción de una base de datos de árboles sintáctico–semánticos para el catalán, euskera y español. In: Proceedings of the XX Conference of the Spanish Society for NLP (SEPLN), Sociedad Española para el Procesamiento del Lenguaje Natural, pp. 81–88 (2004)
8. Taulé, M., Martí, M., Recasens, M.: AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In: Proceedings of 6th International Conference on Language Resources and Evaluation (2008)
9. McDonald, R., Lerman, K., Pereira, F.: Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X), pp. 216–220 (2006)
10. Nivre, J., Hall, J., Nilsson, J., Eryiğit, G., Marinov, S.: Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X), pp. 221–225 (2006)

Advanced Searching in the Valency Lexicons Using PML-TQ Search Engine[⋆]

Eduard Bejček, Václava Kettnerová, and Markéta Lopatková

Charles University in Prague, Institute of Formal and Applied Linguistics
{bejcek,kettnerova,lopatkova}@ufal.mff.cuni.cz

Abstract. This paper presents a sophisticated way to search valency lexicons. We provide a visualization of lexicons with such built-in searching that allows users to draw sophisticated queries in a graphical mode. We exploit the PML-TQ, a query language based on the tree editor TrEd. For demonstration purposes, we focus on VALLEX and PDT-VALLEX, two Czech valency lexicons of verbs. We propose a common lexicon data format supported by PML-TQ. This format offers easy viewing both lexicons, parallel searching and interlinking them. The proposed method is universal and can be used for other hierarchically structured lexicons.

Keywords: Valency lexicon, searching, browsing, linking lexicons.

1 Motivation

Valency lexicons play a crucial role in modern theoretical and computational linguistics. The richer information they provide, the more sophisticated tools are needed for using them, namely for searching them. Search and visualization tools allow users to get useful information from the lexicons, not only for the purposes of theoretical study, but also for the lexicographical aims, e.g., providing frequency analysis, modifying annotation schemes of the lexicon, or consistency checking. However, on-line versions of current valency lexicons do not commonly allow a researcher to ask more complicated, complex queries.

In this paper, we introduce a sophisticated way to search valency lexicons. For demonstration purposes, we focus on VALLEX and PDT-VALLEX, two Czech valency lexicons of verbs. These lexicons represent a collection of manually linguistically annotated data resulting from an attempt at a formal description of valency frames of Czech verbs. Both lexicons are closely related to the Prague Dependency Treebank (PDT henceforth [1,2]) but they capture slightly different types of information and their data structures are different.

The work presented here has two goals:

1. to transform valency lexicons into a common format; this allows a user to search the lexicons in a parallel way and thus facilitate their interlinking at the level of lexical units in the future and

[⋆] This work has been supported by projects MSM 0021620838 and LC536 of the Ministry of Education and grants GAUK 4200/2009 and partially GAUK 52408/2008 of the Grant Agency of Charles University in Prague.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 51–58, 2010.

2. to provide visualization of VALLEX and PDT-VALLEX with such built-in searching that allows users to formulate complex queries in a user-friendly way by drawing their graphical representation.

We exploit the PML-TQ, a query language and search engine designed for querying annotated linguistic data [3], which is based on the TrEd toolkit [4]. There are three important reasons for adopting TrEd and PML-TQ: (i) PML-TQ incorporates a powerful query language useful for complex data and offers graphical query representation, (ii) tree editor TrEd provides us with customizable visualization of richly structured data and makes it possible to visualize query results as well, and (iii) TrEd data format proved to be a suitable common representation for both lexicons and for links between them.

As the PDT-VALLEX lexicon (a part of PDT 2.0 data) can be already searched using PML-TQ we transform the VALLEX lexicon into the format supported by this search engine.

Related Work. Let us mention some of the lexicons providing valency information and their searching interfaces.

More than 960 semantic frames can be browsed in *FrameNet* given a name of the frame or a lemma to search for; in addition, FrameGrapher visualizes relations between semantic frames and their frame elements.[1] The *VerbNet* project maps PropBank verb types to their corresponding Levin classes; on-line search tool facilitates searching only for verb lemmas; VerbNet viewer 'Inspector' can parse a VerbNet data file and print specified attributes for classes.[2] Project *SemLink* combines four lexical resources, PropBank, VerbNet, FrameNet, and WordNet; it supports lemma and semantic class on-line search through Unified Verb Index (UVI).[3] The *Corpus Pattern Analysis* shows patterns with which a verb is associated; it can be browsed only for a given lemma.[4] Verbs in another Czech valency lexicon, *VerbaLex*, can be sorted (similarly to VALLEX) by alphabet, verb roles, morphemic forms, verb classes etc.[5]

Some of these lexicons are already interlinked, like UVI for English (interlinked on the level of individual lexical units). Our long-term goal is to link the VALLEX and PDT-VALLEX lexicons on the level of individual lexical units.

Although our current effort focuses only on searching the VALLEX and PDT-VALLEX lexicons, the underlying search engine can be easily adopted for any other lexicons with structured lexical entries.

2 Two Valency Lexicons

In this section, we provide a basic description of the valency lexicons of Czech verbs, PDT-VALLEX and VALLEX. Both these lexicons take the Functional Generative Description (FGD [9]) as their theoretical background. In Section 2.1, we focus on the differences between their data formats.

[1] http://framenet.icsi.berkeley.edu/FrameGrapher/ [5]

[2] http://verbs.colorado.edu/verb-index/vn/reference.php [6]

[3] http://verbs.colorado.edu/semlink/

[4] http://deb.fi.muni.cz/pdev/ [7]

[5] http://nlp.fi.muni.cz/verbalex/htmlDEMO/ [8]

PDT-VALLEX. PDT-VALLEX (see esp. [10,11]) stores the information on the valency frames of Czech verbs (and also of some nouns, adjectives, and adverbs), which occur at least once in PDT 2.0. Valency frames in PDT-VALLEX are linked with the occurrences of verbs in PDT 2.0. One of the main purposes of building PDT-VALLEX was to ensure the data consistency of PDT.

VALLEX 2.5. The VALLEX lexicon (see esp. [12]) aims at describing valency behavior of verbs in each of their senses, i.e., at providing analysis of whole verb lexemes. In addition to valency frames, further syntactic information is rendered there, esp. the information related to the surface manifestation of verbal valency (e.g., reciprocity, reflexivity, grammatical control), and syntactico-semantic class for a substantial subset of verbs.

In VALLEX, the concept of a *lexeme* plays a crucial role – aspectual counterparts[6] are treated within a single lexeme, which may be therefore represented by more than one lemma. Moreover, a particular lemma may have different orthographic variants. A lexeme associates individual *lexical units* (LUs) representing different verb meanings. The concept of lexeme can be exemplified by the verbs *započítávat*^{impf} and *započítat*^{pf} ‘to count’ as aspectual counterparts and the verb *započíst*^{pf} as an orthographic variant of the verb *započítat*^{pf}. In VALLEX, all these verbs are treated within one lexeme (Figure 1, left column). Let us mention at least the most important reasons for such convention:

- theoretical adequacy: aspectual counterparts have (in principle) the same meaning; in the FGD theory, they are considered as different forms of one verb lexeme;
- compact representation: aspectual counterparts prototypically share the set of lexical units describing their valency characteristics, see LU1 in Figure 1; thus this representation effectively reduce the redundant information in the lexicon;
- convenience for human users when searching the lexicon.

2.1 Data Formats

Although both VALLEX and PDT-VALLEX are stored in XML format, their data formats differ as the lexicons are developed separately and they contain slightly different types of information. In this section, we discuss and exemplify these differences in more detail. The full format description can be found in [13] for the VALLEX format and [14, Section 6.2] for the PDT-VALLEX format.

VALLEX format. VALLEX, version 2.5, is stored in a complex format that reflects the concept of lexeme associating aspectual counterparts of verbs.

The complicated XML format makes searching in VALLEX format rather difficult from the technical point of view: for each lemma, it is necessary to identify correctly the relevant XML elements. For instance, some lexical units are ascribed to all lemmas, see LU1, whereas others are assigned exclusively to some of them, see LU2 ascribed only to the lemma *započítat*^{pf} (Figure 1).

[6] Roughly speaking, perfective and imperfective aspectual counterparts are verbs with the same meaning, which differs in presenting the event either as completed, or as ongoing, like e.g. *pokrýt*^{pf} and *pokrývat*^{impf} as in ‘he covered the floor with the carpet’ and ‘he was covering the floor with the carpet’. Aspectual counterparts usually form pairs, but also triples or even quadruples may appear.

PDT-VALLEX format. In PDT-VALLEX, neither the aspectual counterparts nor the orthographic variants are clustered together. Therefore the XML format of the lexicon is much simpler in comparison with the format of VALLEX. E.g., unlike VALLEX, the verbs *započítávat*, *započítat*, and *započíst* ‘to count’ are described by three separate word entries in PDT-VALLEX. As a result, these word entries are characterized by the identical set of valency frames.

To facilitate comparing, parallel viewing, and interlinking the VALLEX and PDT-VALLEX data, it is necessary to have a common data representation. This representation must be powerful enough to store different type of information from both lexicons. For this purposes, we exploit TrEd toolkit and its native PML format, which is a part of the pmltq extension to the tree editor TrEd.

3 Exploiting TrEd Toolkit for Valency Lexicons

The Tree Editor (TrEd) is a graphical editor that was primarily designed as an annotation tool for the syntactic annotation of the PDT. However, the editor can also be used for data viewing and for advanced data searching. TrEd supports any tree-like structure (which every XML exactly is), thus it is possible to use it for our valency lexicons as well.

TrEd supports an XML-based format called PML (Prague Markup Language [15]). PML data are described in a form of a PML-schema (similarly as DTD describes XML data). In principle, a PML-schema can be obtained automatically from a DTD for an XML document. In addition, it is necessary to specify ‘PML roles’. These roles identify XML elements that ought to be visualized, those XML elements that serve as tree nodes etc. Lastly, there is a stylesheet in TrEd that defines the style, color and layout of nodes, edges and labels of a tree.

The PML data format makes it possible to exploit the PML Tree Query language (PML-TQ[3]), a search tool designed for linguistically annotated data in PML format.

3.1 Common Lexicon Format `vallex_pml`

The first task is to prepare common representation of both lexicons. We use the PML format derived from the PDT-VALLEX format; it is referred to as `vallex_pml` here.

VALLEX format to `vallex_pml`. To transform VALLEX 2.5 data into the `vallex_pml` format, each lexeme from VALLEX have to be split – separate *verb entries* must be created for each lemma. To each verb entry, an appropriate set of *frame entries* describing valency frames (and some other syntactic information) has to be assigned, see Figure 1.

However, the information on aspectual counterparts and their corresponding valency frames belong to the core information stored in the lexicon. To retain this information, the resulting verb entries and corresponding frame entries are interlinked in the target format: each verb entry contains a reference to relevant verb entries for aspectual counterpart(s) (if applicable); similarly, each frame entry contains a reference to corresponding frame entries, see Figure 1. Orthographic variants of lemmas are treated in the same way.

By this splitting and linking, we overcome the aforementioned difficulty with searching in the complex VALLEX format.

PDT-VALLEX format to `vallex_pml`. Though PML is not a native format for PDT-VALLEX, it can be straightforwardly transformed into this format. As a result, PDT-VALLEX is compatible with PML-TQ and TrEd toolkit.

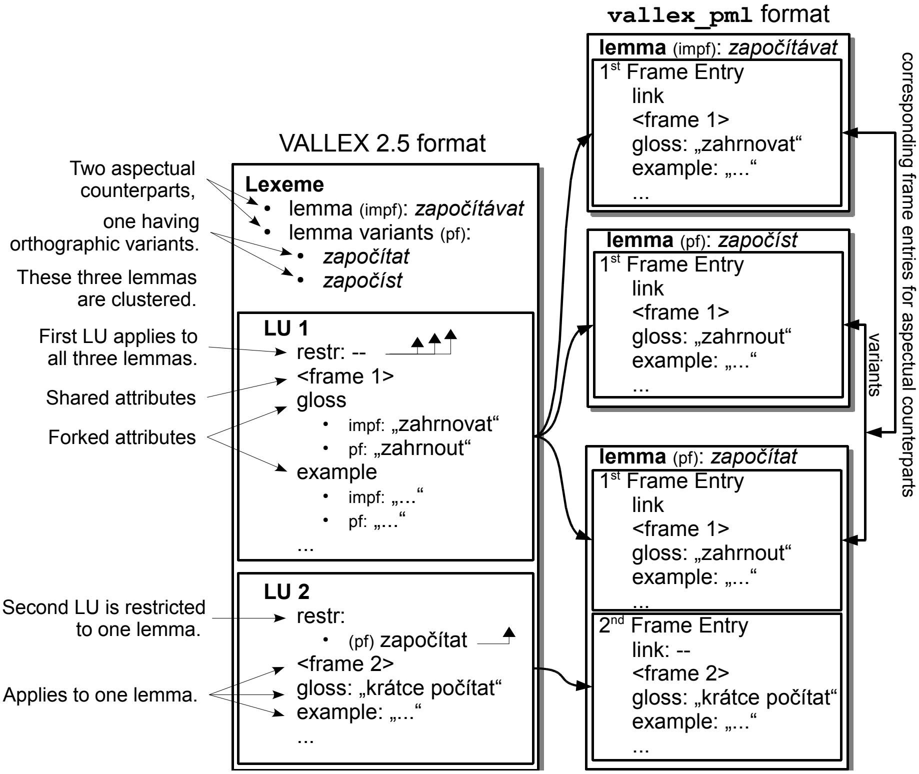


Fig. 1. The transformation of VALLEX into the `vallex_pml` format. The lexeme represented in VALLEX 2.5 by the lemmas *započítávat*^{impf}, *započítat/započíst*^{pf} ‘to count’ associated with two lexical units LU1 and LU2 is schematically displayed in the left column. The right column shows three verb entries for these lemmas and the relevant frame entries for each of these lemmas in the `vallex_pml` format.

After converting both valency lexicons into `vallex_pml` format, they can be loaded into TrEd and three tasks are easier to process: viewing and editing both lexicons (3.2), parallel searching the lexicons (3.3), and linking them together (4).

3.2 Viewing and Browsing in TrEd

Both VALLEX and PDT-VALLEX can be displayed in TrEd as it is shown in Figure 2 (the style depends on the provided PML-schema and the stylesheet).

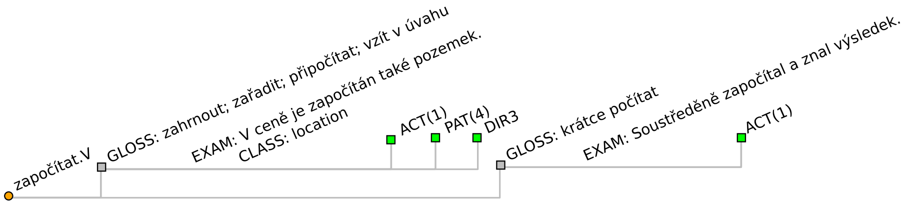

započítat.V
GLOSS: zahrnout; zařadit; připočítat; vzít v úvahu
EXAM: V ceně je započítán také pozemek.
CLASS: location
ACT(1)
PAT(4)
DIR3
GLOSS: krátce počítat
EXAM: Soustředěně započítal a znal výsledek.
ACT(1)

Fig. 2. The verb *započítat* 'to count' displayed in TrEd. The left node represents the lemma of the verb, its two children are two frame entries with different meanings. Both are provided with a gloss and an example. The upper level of nodes represents individual valency complementations and their possible possible morphological forms.

Furthermore, in VALLEX, each member of aspectual group displays the reference(s) to its counterpart(s). By clicking on the reference, the corresponding frame entry (or the whole corresponding verb entry) is displayed in a new window. VALLEX and PDT-VALLEX are interlinked by similar reference on the level of verb entries.

Let us anticipate that the links between frame entries (which correspond to individual meaning of verbs) across the lexicons can be viewed in the same way, see Section 4.

3.3 Searching the Lexicons Using PML-TQ

The lexicons can be not only viewed but also searched using PML Tree Query language. TrEd with PML-TQ extension allows users to formulate complex queries in a user-friendly way. All queries can be created in a graphical mode, a query having a form of a subtree with possible constraints on nodes and edges. Graphical interface enables users to insert nodes into a query subtree, to interconnect the nodes and to formulate constraints on their attributes. (Alternatively, a textual form of the query can be used.)

Let us exemplify some types of possible queries in PML-TQ. There are simple queries, as e.g., 'search for verbs with obligatory ADDRESSEE'. Queries with quantification can be asked too, e.g., 'find a verb with more than twenty valency frames'. Moreover, PML-TQ makes it possible to formulate complex queries concerning diverse properties of verbs, as e.g. the query in Figure 3. We can also search the previous queries in both lexicons in a parallel way.

The output from PML-TQ can be either just viewed in TrEd lemma by lemma, or it can be further processed – one can, for instance, ask for statistics (as, e.g., 'display

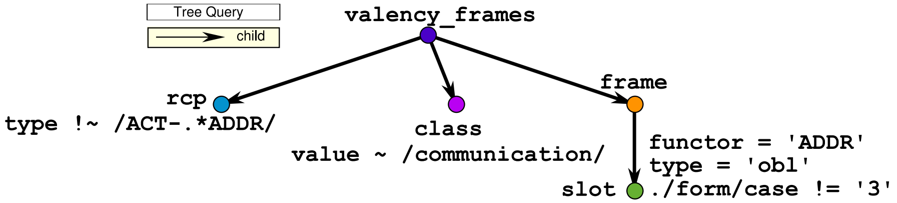


Fig. 3. Example PML-TQ query: searching for verbs from the class 'communication' with obligatory ADDRESSEE realized in other than dative case that cannot be in a reciprocity relation with ACTOR

frequency of found lemmas in individual verb classes'). This is achieved by 'filters' that can be appended to every query and that generate simple text tables.

3.4 Other XML-Based Lexicons

Czech valency lexicons serve here only as sample lexicons allowing us to demonstrate pros and cons of proposed common representation of lexicons. In fact, any XML-based[7] lexicon can be – after some necessary modifications – viewed in TrEd; however, hierarchical, highly structured lexicons benefit from this format most. First necessary step consists in creating a PML-schema; this includes automated transformation from DTD and manual assignment of a few PML roles (they create the required tree structure). Secondly, stylesheet for viewing the lexicon in the required way must be specified. It includes layout of displayed lexicon elements, their descriptions, colors etc. Thirdly, optionally, lexicon data can be transferred into the database for faster querying.

It is profitable not only for our example lexicons VALLEX and PDT-VALLEX but also for e.g. VerbaLex, the other Czech valency lexicon. Its format is different but captures similarly structured information and would be easily transformed to PML.

4 Conclusion and Future Work

In this paper we have presented a format for linking valency lexicons and an effective way how to visualize them in the tree editor TrEd. We have exploited a powerful PML-TQ search engine offering a graphical query representation for comfortable work of linguists. These tools can be used for any lexicon after transformation to PML format, which is mostly automated. It is especially profitable for lexicons with a hierarchical structure such as our example lexicon VALLEX (with several levels of lemma clustering) or as PDT-VALLEX (with structured frame slot information).

The `vallex_pml` format introduced here proved to be suitable common representation for these lexicons. This new data format, which overcomes different logical structures of VALLEX and PDT-VALLEX, poses an important prerequisite for interlinking both valency lexicons – more precisely, for (semi)automatic interlinking corresponding lexical units from VALLEX and PDT-VALLEX – and thus making available information stored in both lexicons (including references to external language resources). This represents an effective way of enriching particular lexical resources.

The `vallex_pml` format being supported by the tree editor TrEd offers parallel visualization of VALLEX and PDT-VALLEX and thus facilitates manual checking and necessary follow-up corrections of the automatic phase of interlinking the affected lexicons as well as viewing and searching in the interlinked system in the future.

References

1. Hajič, J.: Complex Corpus Annotation: The Prague Dependency Treebank. Veda, Bratislava, Bratislava, Slovakia, pp. 54–73 (2006)
2. Hajič, J., et al.: Prague Dependency Treebank 2.0. Linguistic Data Consortium, Philadelphia (2006)

[7] Naturally, other format can be used, too, yet the transformation into PML is not automated.

3. Pajas, P., Štěpánek, J.: System for Querying Syntactically Annotated Corpora. In: Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, Suntec, Singapore, pp. 33–36. Association for Computational Linguistics (2009)
4. Pajas, P., Štěpánek, J.: Recent Advances in a Feature-Rich Framework for Treebank Annotation. In: Scott, D., Uszkoreit, H. (eds.) Proceedings of The 22nd International Conference on Computational Linguistics, The Coling 2008 Organizing Committee, Manchester, UK, vol. 2, pp. 673–680 (2008)
5. Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended Theory and Practice (2006) `http://framenet.icsi.berkeley.edu/book/book.html`
6. Kipper, K., Korhonena, A., Ryant, N., Palmer, M.: Extending VerbNet with Novel Verb Classes. In: Fifth International Conference on Language Resources and Evaluation, LREC 2006 (2006)
7. Hanks, P.: Mapping meaning onto use: a Pattern Dictionary of English Verbs. In: AACL 2008 (2008)
8. Hlaváčková, D., Horák, A.: VerbaLex – New Comprehensive Lexicon of Verb Valencies for Czech. In: Computer Treatment of Slavic and East European Languages, Bratislava, Slovakia, Slovenský národní korpus, pp. 107–115 (2006)
9. Panevová, J.: Valency Frames and the Meaning of the Sentence. In: Luelsdorff, P.A. (ed.) The Prague School of Structural and Functional Linguistics, pp. 223–243. John Benjamins Publishing Company, Amsterdam (1994)
10. Hajič, J., et al.: PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In: Nivre, J., Hinrichs, E. (eds.) Proceedings of The Second Workshop on Treebanks and Linguistic Theories, pp. 57–68. Vaxjo University Press, Vaxjo (2003)
11. Urešová, Z.: The Verbal Valency in the Prague Dependency Treebank from the Annotator's Point of View. In: Šimková, M. (ed.) Insight into Slovak and Czech Corpus Linguistics, Veda, Bratislava, pp. 93–112 (2006)
12. Žabokrtský, Z., Lopatková, M.: Valency Information in VALLEX 2.0: Logical Structure of the Lexicon. Prague Bulletin of Mathematical Linguistics, 41–60 (2007)
13. Žabokrtský, Z.: Valency Lexicon of Czech Verbs. Ph.D. thesis, ÚFAL MFF UK, Prague, Czech Republic (2005)
14. Mikulová, M., et al.: Annotation on the Tectogrammatical Level in the Prague Dependency Treebank. Annotation Manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep. (2006)
15. Pajas, P., Štěpánek, J.: A Generic XML-Based Format for Structured Linguistic Annotation and Its Application to Prague Dependency Treebank 2.0. Technical Report TR-2005-29, ÚFAL MFF UK, Prague, Czech Rep. (2005)

These Nouns That Hide Events: An Initial Detection

Amaria Adila Bouabdallah, Tassadit Amghar, and Bernard Levrat

LERIA - University of Angers, 2 Bd Lavoisier 49000 Angers, France
{amaria,amghar,levrat}@info.univ-angers.fr
http://www.info.univ-angers.fr/leria/

Abstract. Many studies have been devoted to the temporal analysis of texts, and more precisely to the tagging of temporal entities and relations occurring in texts. Among these lasts, the various avatars of events in their multiples occurring forms has been tackled by numerous works. We describe here a method for the detection of noun phrases denoting events. Our approach is based on the implementation of a simple linguistic test proposed by linguists for this task. Our method is applied on two different corpuses; the first is composed of newspaper articles and the second, a much larger one, rests on an interface for automatically querying the Yahoo search engine. Primary results are encouraging and increasing the size of the learning corpus should allow for a real statistical validation of the results.

Keywords: Nominal event recognition, temporal annotation.

1 Introduction

The detection of temporal information is a crucial task for automatic text processing. It is not only used in linguistics for the modelization of linguistic phenomenon and reasoning implying time entities [1] but also in numerous applications in language comprehension, information retrieval, question-answering and information extraction. In that way, many ontologies were developped, to reflect different aspects of temporal phenomena expressed by the nearly natural languages [2,3,4,5]. Have also emerged many tools of detection and standardization of such data leading the improvement of performance and coverage of applications involving time [6] such that many systems of question-answering and information extraction. Many of the works devoted to the temporal analysis of documents, many of them are particularly interested in the identification and annotation of different types of temporal expressions and events. Among the work on the identification of temporal entities, [7] which deals only on absolute temporal entities (dates, for example), in [8] authors worked on the semantic markup calendar entities time, or more recently [9] developping a functional analysis for formal representation of calendar expressions in the context of a navigation system for biographical texts. Some other works are mainly interrested in the identification and annotation of events: [6] and [10], which are based on a specification language TimeML[1] [11] for the annotation of temporal information.

Contrary to our notion of event defined later in Section 2.2, TimeML expands the class of events to include also some states. This view corresponds to the notion of "eventuality" [12], in their various grammaticals forms: verbs, nouns and adjectives.

[1] From the workshop TERQAS (Time and Event Recognition for Question Answering Systems) in the project AQUAINT.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 59–66, 2010.

In this paper, we propose a statistical approach to detect events described by noun phrases in a corpus on newspaper articles written in French.

2 Aspect in Verbal Domain

2.1 Classification of Verbs

Classifications established in [13] or in [14], split verbs (or verb sentences) into four classes: (1) states, (2) activities, (3) accomplishments and (4) achievements. These four classes can be intuitively characterized as follows:

1. **States:** process with neither start and nor end, they include no change idea. They last a time but do not address the idea of a process that evolves over time [15]. We cannot ask how long they took, or if they ended, and are not compatible with a description as a point of time. For example: *Paul connaît le fonctionnement du systême* (Paul knows the functioning of the system).

   Verbs called psychological, as *penser* (to think), *savoir* (to know) and *croire* (to believe), describe states.

   Thus, the sentence *Jean pense que la terre est ronde* (Jean thinks that the earth is round) denote the description of a state.

   Similarly, the intentional verbs such as *vouloir* (to want), *désirer* (to desire) and *souhaiter* (to wish), denote psychological states. It is also the case of verbs like *aimer*(to love), *adorer* (to adore) and *détester* (to detest).

   The verbs of possession *avoir* (to have), *posséder* (to possess), and *appartenir* (to belong) as well as the verbs *habiter* (to live), *rester* (to remain) and *exister* (to exist) denote states because they involve no change. They are stative verbs.
2. **Activities:** are associated with an idea of progression over time, with a possible measurement of this progress, but without expected term.

   Stopping the activity at any time after a minimum time that exceeds does not affect the achievement of this process. Thus, we can characterize it as a process without a determined end, with a durative nature, not conclusive. For example: *Paul programme en lisp*(Paul programs Lisp).

   When the object of activity is expressed, more precisely, when the purpose of the act is expressed, it becomes an accomplishment.
3. **Accomplishments:** This dynamic class progresses over time towards an end. Only the the achievement of this term can achieve the performance and makes it conclusive. For example: *Paul programme l'algorithme de Robinson.* (Paul is programming Robinson's algorithm).

   We can see the difference between these last two classes Activities/Accomplishments by the folowing way: assuming that Paul had to be interrupted in the case of the activity, we can still conclude that Paul has programmed in Lisp, but we cannot say that Paul programmed the algorithm of Robinson.
4. **Achievements:** This type of process describes the beginning or the ending of an action and results in a single point. This class concerns essentially psychological and perception verbs. These process are essentially characterized by their punctual character. For example: *Paul a perdu son mot de passe* (Paul lost his password).

> According to certain authors, such [16] and [17], accomplishments and achievements may be grouped together under the term of event.

It is commonly admitted that two main criteria of distinction of these various categories are i) the dynamicity and ii) telicity [18] – see Table 1.

Table 1. Criteria of distinction of the differents categories

	Dynamicity	Telicity
States	−	−
Activities	+	−
Acc & Ach	+	+

2.2 The Notion of Event

Bach [12] groups events and stats together under the term *eventuality*. Events have spatiotemporal properties and causal effects, they are localizable in the world, because of the participants in their predicative structure: an agent who activates them at a given moment and possibly a theme/object which undergoes them. Events are distinguished from stats in that they result in changes of the stat of the world, (*i.e.* they have causal effects). Thus, events group together activities, accomplishments and achievements previously quoted.

To characterize the events, [19] depicts a study concerning the context of appearance and their uses in English. [20] sketched a similar work for French. Both cases, borrow the notion of container of Zeno Vendler [14]. This later notion consists in the demarcation of the linguistic contexts and the constructions which require the presence of an event. In French, for events, verbs are *arriver* (to arrive), *se produire* (to occur), *avoir lieu* (to take place), *se passer* (to happen)... which all take an event, necessarily under its nominal form, in subject position. For example: *La chute du mur de Berlin est arrivée/s'est produite/a eu lieu hier* (The fall of the Berlin Wall arrived / occurred / took place yesterday).

Verbs such as *assister à* (to attend), *être témoin de* (to witness of), and *manquer* (to fail) need for an event object, occuring also under nominal form. For example: *Jean a assisté à la chute du mur de Berlin* (Jean attended the fall of the Berlin Wall).

Segmented relative constructions (For example: *Ce qui s'est passé, c'est que le mur de Berlin est tombé* (What happened is that the Berlin Wall fell)), or (For example: *C'est en 1989 que le mur de Berlin est tombé* (It is in 1989 when the Berlin Wall fell)) and impersonal (For example: *Il s'est passé que le mur de Berlin est tombé* (It happened that the Berlin Wall fell)) with verbs as *arriver* (arrive), *se produire* (occur)... are also containers which require the intervention of an event, but in these cases, the event argument is under the form of an object clause, introduced by the subordinating conjunction *que* (that).

The notion of event container of Vendler, already we evoked above, groups together the verbs which require an event subject (*se passer* (to happen), *se produire* (to occur),*arriver* (to arrive) and *avoir lieu* (to take place)), or an event object (*assister à* (to attend), *être témoin de* (to witness of), and *manquer* (to fail)). Arguments of these verbs have to be either in nominal form, *la chute du mur de Berlin a eu lieu* (The fall of the Berlin Wall took place), or pronominal, *ça a eu lieu* (That took place). The container determines the type of entity which a pronoun can refer.

3 The Aspect in the Nominal Domain

In [18], authors ask the question of the application to noun phrases: from the aspectual criterion of the dynamicity, we can conclude that the dynamic aspect exists in the nominal domain, although it is not always easy to bring to light, and the aspectual inheritance of the dynamicity is not observed for all the deverbal nouns. More exactly, when the name has a concrete interpretation, it is not endowed with aspectuals features and more over may possibly express the dynamic aspectual value. The following working hypothesis was formulated: when a nominalization has aspectuals characteristics, then it preserves the specification of the [+ or − dynamics] features of its verbal base. In other words, the deverbal nouns of states, activities, achievements and accomplishments, in their abstract meaning, agree with static/dynamic feature of the verbs they are coming from. The criterion of dynamicity, allows to distinguish static from dynamic nouns, and therefore distinguish between stats of three other classes: activities, achievements and accomplishments are grouped in the class of events, previously defined.

4 Description of the Corpus

Our corpus extracted from complete editions of the regional (daily paper): *L'est républicain*.

These texts are quite rich in expressions denoting events under various forms. Our study bears on 358 journalistic articles from 2002. These articles are represented under a XML format and are subjected to a pretreatment allowing to extract the textual contents from it.

5 Detection of Event Nouns

We implemented a linguistic test for the detection of event nouns by a distributional statistical treatment. To do it, we use UNITEX[2]: a platform used to manipulate a finite state machines adapted to the NLP (Natural Languages Processing). It allows to describe linguistic patterns in the form of regular expressions or automaton to find their locations in a text. In first step, we locate the noun phrases of the text by means of automaton equiavalent to regular expressions presented in Table 2.

[2] `http://www-igm.univ-mlv.fr/~unitex`

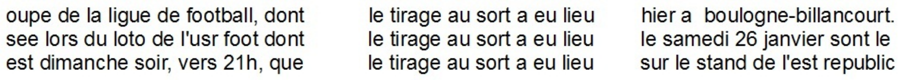
Table 2. Regular expressions for the location of noun phrases

Singular	$< DET : s > .(< A : s > + < E >). < N : s >$
Plural	$< DET : p > .(< A : p > + < E >). < N : p >$

The first expression corresponding to the singular form, it is the concatenation of three elements, $< DET >$ corresponds to determiners preceding nouns, $(< A : s > + < E >)$ corresponds, either in the singular adjectives, or in the empty chaines and $< N : s >$ corresponds to the singular nouns. For example, in the sentence extracted from our entry text "Le jeune couple s'est installé à Paris" (The young couple moved to Paris), the first regular exepression allows to locate the constituent "Le jeune couple" (young couple).

5.1 The Characterization of Events

Let us remind that the events are spatiotemporal entities, having causal and localizable effects in the world. They may arrive, occur or, more generally, take place. We implement a linguistic test proposed by [18] to locate the event nouns: "*Noun denoting events can appear in position of N in* ***N took place*** *(in such place, at such moment), describe entities endowed with a temporal demarcation*". For example, "L'explosion a eu lieu" (The explosion took place). We developed a function, allowing to look for all the concordances formed by the nouns of the text in entry already located previously with the constituent "a eu lieu" (took place) in our corpus. For each occurrence of "GN a eu lieu" (Nominal Phrase took place), the function calculates a left context and a right context, the size of which was fixed to 40 characters of every side (see Figure 1).



```
oupe de la ligue de football, dont     le tirage au sort a eu lieu    hier a  boulogne-billancourt.
see lors du loto de l'usr foot dont    le tirage au sort a eu lieu    le samedi 26 janvier sont le
est dimanche soir, vers 21h, que       le tirage au sort a eu lieu    sur le stand de l'est republic
```

Fig. 1. Extract of the concordance of "the drawong lots took place" in the corpus

To eliminate "false positives", we eliminate all the concordances with a left context ends by prepositions, such as: de (of), par (by), sur (on).... Indeed, in these cases, the noun associated with "a eu lieu" (took place) does not represent the main subject of what occurred, but a complement noun. This last one represents the subject of the action which took place well. For example, the left context of the concordance:

"l'audition des eleves de **l'école de musique a eu lieu** à la salle des fêtes"

ends by the preposition "de" (of) and we show that the main subject of the action is "l'audition des élèves" (The audition of the students) and not "l'école de musique" (Music School). After the elimination of this type of concordances, we obtain a file containing all the concordances with nouns preceding by "a eu lieu" (took place) denote event nouns.

6 Evaluation

As a first evaluation, we selected randomly a text from our corpus. Our evaluation aims to estimate in term of recall and precision, the capacity of the method to detect the event nouns located in this text. *Recall* is defined as the number of the correct event nouns detected by our method devided by the total number of the nominal events in the text. *Precision*, is defined as the percentage of the correct event nouns detected devided by the total number of events detected by the method. The text used has been manually tagged. From the selected text, a number of 464 nouns were located by the first stage of our method. From these nouns, 52 nouns were manually tagged as events. Our method detected 10 nouns as "nominal events", among which 10 nouns, 9 are correct. These results lead to precision rise to 90% and recall to 17.30%.

This results presented above allow to draw following conclusions. Recall requires to be greatly improve, the simplicit way to achieve this, is to increase the size of corpus. Obviously implementing more sophisitcated linguistic tests for detecting events, will also imporve recall. For example, accepting synonyms of the pettern "a eu lieu" (took place) such as: "s'est produit" (occured), "est arrivé" (arrived)...

The improvement of precision can be achieved by increasing the contextual elements taken into account in the filtering, for example, the occurrence of the temporal adverbial expression preceding "a eu lieu" (took place). For example:

"et les axes ruraux. **Ce week-end a eu lieu** une phase de montée en"
(and the rural axes. **This week-end took place** an increase phase in) As shows in the example, "Ce week-end" (This weekend) does not denote an event nouns, but an adverb of time.

To improve the recall, we applied our method to all the web pages in french from Yahoo. We develop a script using the Application Programming Interface[3] to question automatically the search engine Yahoo. From the same list of 464 nouns of the entry text used previously, we built 464 queries. Each query consists of a noun combined with "a eu lieu" (took place). For example, for the noun phrase "la montée des eaux" (the water rise), the query will be: "la montée des eaux a eu lieu" (the water rise took place). The API Yahoo also allows, for every found document, to extract the part of the text containing the query in question. This allows afterward to re do a treatment previously made on the corpus. In particular, the elimination of all the concordances with a left context ended by prepositions (to see Section 5.1). On 464 nouns taken in entry, 64 nouns were tagged as events since:

- Number of documents returned is greater than 0.
- The left context preceding the constituent "Nom a eu lieu" (noun took place) in the returned texts, does not end by a preposition.

On 64 event nouns, 47 nouns correspond to the events labelled manually ($N = 52$), while 17 do not represent events. Results leads in a recall of 90.38%, and a precision of 73.44%. On the other hand, we also notice that in spite of the decrease of the precision which is crossed 90% in 73.44%, this last one remains relatively high. Other problem using well do extended the corpus, leads to cup with the various types of coding used in web pages, unlike a corpus, where we use a single type of coding.

[3] http://developer.yahoo.net

7 Conclusion

We proposed in this article, a method for the location of the event nouns in French. Our work uses two stages. The first one consists in detecting all the nouns of a file in entry, by using UNITEX. Then these nouns, are filtred to keep only nouns denoting events. To do this, we are based on a pattern search in our corpus. Corpuses in interest are those when time entities are noticable, for example, articles of press, bibliographic texts or still narrative texts. Our results reflect well the interest of the method, especially with a high precision. On the other hand to improve the rate of recall, we intend to increase the size of our corpus, and consequently, to increase the chances to find the best patterns which serve as base for the detection of all the event nouns, as the fact appear in the second evaluation applied to a wider corpus.

Other way will be tested to recorgnize the event nouns, such as:

1. presence of arguments and especially time arguments, like: “o'clock”, “today”
2. presence of morphological affixes used to detect nouns such as “ion” in the example of “explosion”.

References

1. Amghar, T., Battistelli, D., Charnois, T.: Reasonning on Aspectual-Temporal Information in French within Conceptual Graphs, p. 315. IEEE Computer Society, Los Alamitos (2002)
2. Gayral, F., Grandemange, P.: Une ontologie du temps pour le langage naturel. In: COLING, pp. 295–302 (1992)
3. Reichenbach, H.: Elements of Symbolic Logic. University of California Press, Berkeley (1947)
4. Mokhtari, A., Kayser, D.: Time in a Causal Theory. In: TIME, pp. 14–20 (1996)
5. Battistelli, D., Desclés, J.: Modalités d'action et raisonnements aspecto-temporels. In: Actes VEXTAL 1999 (Venezia per il trattamento automatico delle lingue), pp. 351–359 (1999)
6. Bittar, A.: Annotation des informations temporelles dans des textes en Français. In: RECITAL (2008)
7. Agency, D.A.R.P.: Proceedings of the Seventh Message Understanding Conferences (MUC-7). Morgan Kaufmann, California (1998)
8. Schilder, F., Habel, C.: From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In: Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing, ACL 2001, Toulouse, pp. 65–72 (2001)
9. Battistelli, D., Minel, J., Schwer, S.: Quelques exemples d'utilisation des S_langages pour le traitement de la temporalité en linguistique. In: SDC Semaine de la Connaissance, SDC 2006, p. 5 (2006)
10. Parent, G., Gagnon, M., Muller, P.: Annotation d'expressions temporelles et d'événements en Français. In: Traitement Automatique des Langues Naturelles (TALN), Avignon. ATALA (2008), http://www.atala.org/
11. Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gauzauskas, R., Setzer, A., Katz, G.: Timeml: Robust Specification of Event and Temporal Expression in Text. In: IWCS-5, Fifth International Workshop on Computational Semantics (2003)
12. Bach, E.: The Algebra of Events. Linguistics and Philosophy 9, 5–16 (1986)
13. Bennett, M., Partee, B.: Towards the Logic of Tense and Aspect in English. Indiana University Linguistic Club (1978)

14. Vendler, Z.: Linguistics in Philosophy. Cornell University Press (1967)
15. Schwer, S., Tovena, L.: Ontologies temporelles et sémantique de la temporalité (2008)
16. Mourelatos, A.: Events, Processes and States. Linguistics and Philosophy 2, 415–434 (1978)
17. Verkuyl, H.: Aspectual Classes and Aspectual Composition. Linguistics and Philosophy 12, 39–94 (1989)
18. Huyghe, R., Marin, R.: L'héritage aspectuel des noms déverbaux en français et en espagnol. Faits de langues 30, 302 (2007)
19. Asher, N.: Reference to Abstract Objects in Discourse: A Philosophical Semantics for Natural Language Metaphysics. SLAP, vol. 50. Kluwer, Dordrecht (1993), http://www.wkap.nl/
20. Amsili, P., Denis, P., Roussarie, L.: Anaphores abstraites en français: représentation formelle. TAL (Traitement Automatique des Langues) 46, 15–39 (2005)

Can Corpus Pattern Analysis Be Used in NLP?

Silvie Cinková[1], Martin Holub[1], Pavel Rychlý[2],
Lenka Smejkalová[1], and Jana Šindlerová[1]

[1] Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
[2] Masaryk University in Brno, Faculty of Informatics, Department of Information Technology

Abstract. Corpus Pattern Analysis (CPA) [1], coined and implemented by Hanks as the Pattern Dictionary of English Verbs (PDEV) [2], appears to be the only deliberate and consistent implementation of Sinclair's concept of Lexical Item [3]. In his theoretical inquiries [4] Hanks hypothesizes that the pattern repository produced by CPA can also support the word sense disambiguation task. Although more than 670 verb entries have already been compiled in PDEV, no systematic evaluation of this ambitious project has been reported yet.

Assuming that the Sinclairian concept of the Lexical Item is correct, we started to closely examine PDEV with its possible NLP application in mind. Our experiments presented in this paper have been performed on a pilot sample of English verbs to provide a first reliable view on whether humans can agree in assigning PDEV patterns to verbs in a corpus. As a conclusion we suggest procedures for future development of PDEV.

1 Corpus Pattern Analysis

1.1 What Is a Lexical Item?

John Sinclair, the Nestor of corpus linguistics, criticized the separation of grammar and lexicon in the sense that the grammar (in extreme cases) only describes the *form* of a lexical item with respect to its potential context, while the lexicon primarily describes the *meaning* comprised by its base form, regardless of the context. Not only are form and meaning tightly related, Sinclair argues [3, p. 59f.], they must even be identical, considering that most ambiguities are resolved by context in authentic language usage. Hence, a description of lexical items should take into account both aspects at the same time.

Instead of describing the paradigmatic properties of each lexical item by listing the potential senses of its lemma, he pleads for describing both the syntagmatic and the paradigmatic properties of each lexical item as patterns in which the given lexical item occurs [3, p. 69].

1.2 Pattern Dictionary of English Verbs (PDEV)

Hanks, Sinclair's collaborator on the first corpus-based dictionary ever, the Collins Cobuild English Language Dictionary [5], has proposed *Corpus Pattern Analysis*

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 67–74, 2010.

(CPA), a semi-formal lexical description method that consistently materializes Sinclair's concept of capturing meanings in patterns of language rather than lexical units in the token-centered lexicographic tradition.

The current CPA captures "normal", i.e. reasonably frequent, usages of a given verb by sorting them into *patterns*. Each pattern is formulated as a proposition in which the verb in question is lemmatized[1] and its relevant collocates are classified by means of two sets of semantic labels or listed as *lexical sets*, depending on whether the respective collocates can be listed (as a lexical set) or grouped together under the general heading of a *Semantic Type*. Each proposition is paraphrased by a sentence in which the relevant pattern arguments are labeled identically with the proposition part. This paraphrase embodies the *implicature* (or *meaning potential*, see [1]) activated by that particular pattern.

Each collocate that cannot be represented by a lexical set, is described by a Semantic Type. Semantic types are sometimes augmented by a *Semantic Role*. The Semantic Types are a finite set of labels hierarchically ordered in what Hanks calls a *shallow semantic ontology* [2]. The Semantic Types describe inherent properties of the collocates, such as *Human*, *Artifact*, *Stuff*, *Document*. The Semantic Roles describe the properties that are assigned to the word in a particular pattern or context.

CPA is implemented as PDEV, the Pattern Dictionary of English Verbs, built by Hanks and his collaborators [2]. It comprises two interlinked components: a list of patterns for each verb and a reference set of manually tagged sample data. Each verb in PDEV is linked to a reference sample of concordances, which contain the verb in question. The sample is randomly selected from the British National Corpus (BNC) [6], and its size is typically 250–500, depending on the semantic complexity of the verb.

We perceive Hanks' patterns as a means of discrimination of Sinclairian lexical items, which, in their own right, imply what is usually referred to as "meaning". To the best of our knowledge, PDEV is the first real and conscious implementation of Sinclair's principles concerning the lexical item and the way it should be described. This fact makes PDEV unique, yet there are certainly a number of other projects that formally describe semantic distinctions of verb uses, with different theoretical foundations, e.g. [7,8].

1.3 PDEV as a Source for NLP?

Hanks' approach to the lexical description of verbs is novel and linguistically sound at the same time. It has gained a world-wide reputation, judging by the more than 600[2] topic-related citations for Hanks, as well as the numerous keynote speeches Hanks has been invited to give on this subject since the first significant mention of CPA in [1]. CPA is intuitively plausible, and its formal encoding appears promising for various applications in NLP – the more so because Hanks has been continuously linking his lexicon to other well-known lexical sources, such as FrameNet [7] or the Erlangen Valency Bank [9].

[1] Exception: passivization.

[2] Harzing's Publish or Perish since Hanks, 1994 (recorded as 1993), quoted 2010-03-24.

However, the “qualified judgment” on the hypothesized NLP usability of CPA pronounced by a number of language experts has not yet been experimentally tested.

With our experiments we are taking a first step towards providing a reliable assessment whether or not the current PDEV is suitable for NLP application. In this short paper we report on an on-going pilot study, in which we examine the consistency of PDEV, which we regard as the basic prerequisite for its NLP-usability. Should we identify problematic issues, we suggest (and plan to implement) improvements based on a pilot sample in the next step.

2 Current Status of PDEV Development

2.1 Platform of PDEV development

The development of PDEV is supported by two interconnected applications. The first one used for pattern editing is based on the “Dictionary Editor and Browser” tool (DEB), a dictionary-making database platform developed at the Masaryk University in Brno (MU), Czech Republic [10]. This platform enables the lexicographic processing of XML-encoded data through a user-friendly web-based graphical user interface integrated as an add-on in the Firefox-Mozilla web browser. The data is stored on DEB servers located at MU. PDEV is one of the numerous applications of DEB. It incorporates a tailored interface for pattern creating, ontology browsing and editing. The second application, used for concordance tagging, is a modified version of the Sketch Engine [11].

2.2 Current PDEV Statistics

PDEV has been developed on basis of verb occurrences in the BNC50 corpus, a 50-million-word part of the BNC. BNC50 contains almost 5,800 verb types occurring in 8 million verb tokens. However, about 41% of all verb tokens represent auxiliary (‘will’, ‘do’, ‘have’, ‘be’) or modal (‘shall’, ‘can’, ‘must’, etc.) verbs that are not analysed in the PDEV project at all. The number of lexical verb types in BNC50 is 5,757 and the total number of the corresponding tokens is 4,673,003. Table 1 illustrates the well known fact that rare words do not significantly contribute to corpus coverage. Verbs with frequency higher than 27 cover the corpus up to 99.5%.

Currently (March 2010) the number of verbs compiled in PDEV is 678, 11.8% of all lexical verb types in BNC50. The number of corresponding tokens in BNC50 is 495,724, which cover 10.6% of all BNC50 lexical verb tokens.

Table 1. The coverage of BNC50 verb tokens. For example, 918 most frequent verbs, each of which occurs at least 610 times in BNC50, cover more than 90% of all BNC50 lexical verb tokens.

min. frequency	54,872	8,723	610	246	136	90	48	28	1
verb types	7	120	918	1,519	2,030	2,452	3,151	3,780	5,757
BNC50 coverage	11%	50%	90%	95%	97%	98%	99%	99.5%	100%

The number of all patterns created for those compiled verbs is 2,572. While the average number of patterns per compiled verb type is 3.79, a more interesting value, the expected number of patterns per token is 9.72 (more frequent verbs often have also more patterns). The correlation between verb frequency and the number of patterns is shown in Fig. 1.

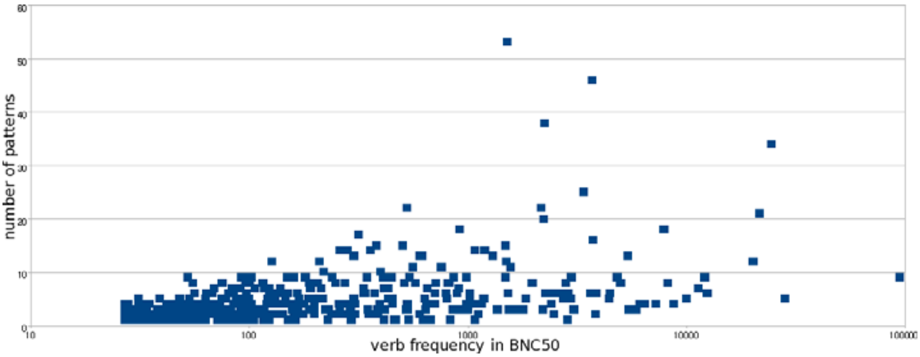


Fig. 1. The number of patterns of 502 PDEV verbs (with frequency at least 28) and their frequency in BNC50

3 First Evaluation of PDEV

3.1 Evaluation Method

PDEV with its tagged reference samples can be regarded as a manually created gold standard data set for machine-learning experiments. So far, the lexicon has mainly been built by Hanks. In terms of annotation, the entire data available has been annotated by one single annotator. Moreover, the author of the patterns and the data annotator are the same person. Our first question was therefore: are humans who did not create the entries themselves able to agree in pattern assignment? A reasonable degree of interannotator agreement is a prerequisite for any further automatic processing.

This assumption has two aspects, which we want to keep apart: creating the lexicon and annotating the data. Here we focus only on the consistency in tagging the data according to already existing patterns. We regard the mutual agreement of independently working annotators as a measure of quality of each given lexical entry.

As with any linguistically rich annotation, the annotators must be clearly instructed and trained before the interannotator agreement can be measured. The authors of this paper, who acted as annotators, have only learned details of the annotation procedure on-the-fly while discussing the patterns as well as own data findings with Hanks, watching him work and having ocassional hands-on experience with creating a new entry for more than one year. No detailed annotation guidelines were available at that point. We expected this fact to lower our inter-annotator agreement. While tagging, we kept each a record of difficult decisions for future reference when a regular annotation guide for new annotators is being created, and we analyzed our records along with the annotated data when all samples were finished. Hanks performed the same annotation with us, and his sample annotation served as a reference in case of doubt.

3.2 Experiments

For the first experiments we chose a pilot sample of 30 verbs selected from the set of complete compiled PDEV verbs. To measure the inter-annotator agreement we used the standard kappa function (Cohen's kappa for annotator pairs, and Fleiss' kappa for more than two annotators). Just for example, results for some of the pilot verbs are shown in Table 2.

Table 2. An example of 6 pilot verbs selected for validation and the interannotator agrement (IAA) measured on random samples selected from two different corpora. PEDT verbs were annotated by only 2 people.

Verbs	Verb Features			IAA on PEDT	IAA on BNC50	
	patterns	perplexity	BNC50 freq.	kappa	annotators	kappa
tell	21	3.80	21,550	0.66	2	0.27
lead	12	3.97	20,180	0.83	3	0.78
call	34	6.68	24,439	0.72	2	0.68
argue	7	1.73	11,362	0.93	3	1
claim	6	3.14	12,517	0.87	4	0.72
fire	15	7.96	1,488	0.71	2	0.42

To compare the inter-annotator agreement on different corpora, we used randomly selected concordance samples both from the BNC50 and from the Prague English Dependency Treebank (PEDT) [12]. PEDT consists of Wall Street Journal articles. The results show that a domain-restricted corpus sometimes implies better inter-annotator agreement. On the other hand, we are aware that patterns designed to fit the BNC50 corpus do not necessarily fit another corpus (the more so a strongly domain-restricted one).

3.3 Disagreement Analysis

We have identified the following types of disagreements:

1. *Vagueness of instructions on context consideration.* Whether, to what extent, and how wide the context is to be taken into account is not yet clearly defined in the theoretical foundations of CPA.
2. *Markable-Unmarkable.* Sometimes it is difficult to decide whether a form is a markable or an unmarkable; e.g. in participial forms.

3. *Elliptical usages.* Elliptical usages are problematic because of their inherent ambiguity. There are two types of ambiguous ellipses: a) the context does not enable a distinction to be made between two potentially relevant patterns, of which one requires zero realization of an argument and the other allows optional omission; or b) there are two patterns with different implicatures, both allowing optional omission of an argument. In such cases, it is not possible to say what the collocation would look like if the ellipsis was restored; so the text meaning can only be determined by examination of the wider context, which is beyond the scope of CPA.
4. *Collocate matches several Semantic Types.* In a few cases, the context fits more than one pattern by its implicature, and one of the pattern-relevant collocates of the verb has several inherent semantic features, of which each allows the collocate to match a different Semantic Type in different patterns.
5. *Insufficient competence in English.* Sometimes, the non-native annotators misunderstood a concordance.
6. *Missing pattern.* The concordances had been taken mainly from BNC, but some tasks contained only concordances from PEDT, which we regard as a domain-restricted corpus. Some usages frequent in PEDT were not explicitly captured by the patterns based on BNC, and the annotators tagged them as exploitations of various different patterns, in which they disagreed. Some (rare) suggestions to add a pattern arose also from the BNC annotation.
7. *Implicatures too fine-grained.* The random sample showed in some cases that the context often does not allow for disambiguation of very fine-grained distinctions between implicatures activated by different patterns.
8. *Semantic Types too fine-grained.* Some (in fact quite numerous) concordances did not match a pattern because the collocates in question did not match the Semantic Type prescribed by the pattern, although intuitively it seemed to fit well with that pattern, too.

3.4 Discussion

The results of the pilot project measuring inter-annotator agreement were not entirely impressive. However, *only a few cases pointed at pattern inadequacy*. Our findings are not too different from the recently published analysis of annotation of polysemous predicates [13].

The annotation procedure had not become routine yet. Many errors were simply oversights: it happened e.g. that an annotator consistently confused one pattern number for another throughout one entire sample, a few concordances were misunderstood and the annotators also sometimes forgot about the fact that one single implicature is split into different patterns when a collocate is typically realized both as a noun or a verb clause (nouns are described by Semantic Types, while verb clauses by syntax), and he/she kept assigning only the one with the Semantic Type to it.

The most frequent type of frame inadequacy that we encountered is easily amended by adding a Semantic Type. We had decided to strictly classify all concordances as “unmarkable”, whenever a concordance intuitively perceived as typical and norm-conforming (i.e. not an “exploitation”) contained a collocate that was not included in the Semantic Types. This happened quite often, since some of the entries analysed had been finished long ago and were compiled with an outdated set of Semantic Types.

Missing/competing patterns, however, were rare, which is a good sign.

4 Prospects and Conclusions

The pilot annotation experiments were conducted to uncover potential problems prior to any large-scale annotation. The experiments helped us specify which issues must be particularly looked into when an annotation manual is being written. Once disagreements caused by points 1, 2, 3 and 5 (Section 3.3) have been eliminated by a better instruction specification and, hopefully, by hiring native speakers, the annotation will provide valuable feedback for pattern building. We suggest the following procedure for the "validation" of lexical entries in PDEV:

1. The initial patterns will be created by Hanks as usual.
2. When declared ready for validation, they will be given to annotators, along with the tagged reference data.
3. The annotators will tag a new randomly selected BNC sample and keep notes on potentially missing patterns, incomprehensible context, etc.
4. The interannotator agreement will be measured, disagreements identified and discussed with Hanks.
5. Based on the disagreement analysis, the patterns will be revised and/or the annotation instructions enhanced.
6. The revised entries will be returned to the annotators, along with a (different) data sample to be tagged.
7. The entire process will be repeated until the inter-annotator agreement (at least in the most relevant points, such as 4, 6 and 7 listed in Section 3.3) has risen to an acceptable level.
8. Each revised lexical entry will be declared as "validated" and ready for machine-learning experiments.

The PDEV patterns seem to be a very promising way of describing verbs as Sinclairian Lexical Items. From the machine-learning view, the pattern inventory and the tagged reference data attached represent two complementary sources, which enable the combination of rule-based and statistical approaches to automatic verb disambiguation. The current PDEV needs increased standardization and regular evaluation of new entries by iterated multiple annotation and interannotator agreement measuring. Our pilot study is a first step towards a systematic validation and building-up PDEV as an NLP-applicable lexical source.

Acknowledgments

This project was kindly supported by EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic), the Czech Science Foundation GA-P406/2010/0875 and the Czech Ministry of Education MSM 0021620838.

This work has been partly supported by the Ministry of Education of CR within the National Research Programme II project 2C06009 and by the Czech Science Foundation under the projects P401/10/0792, 102/09/1842 and 407/07/0679.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement number 248,307.

We would like to thank Patrick Hanks for his comments on an earlier draft of this paper. We also thank three anonymous reviewers for their valuable feedback.

References

1. Hanks, P.: Linguistic Norms and Pragmatic Exploitations, Or Why Lexicographers need Prototype Theory, and Vice Versa. In: Kiefer, F., Kiss, G., Pajzs, J. (eds.) Papers in Computational Lexicography: Complex 1994. Hungarian Academy of Sciences, Budapest (1994)
2. Hanks, P., Pustejovsky, J.: A Pattern Dictionary for Natural Language Processing. Revue Francaise de linguistique appliquée 10(2) (2005)
3. Sinclair, J.: The Lexical Item. In: Hanks, P. (ed.) Lexicology: Critical Concepts in Linguistics, 6 vols. Routledge, London; First published on Weigand, E. (ed.): Contrastive Lexical Semantics Amsterdam, pp. 1–24. John Benjamins, Amsterdam (1998, 2008)
4. Hanks, P.: The Lexicographical Legacy of John Sinclair. International Journal of Lexicography 21(3) (2008)
5. Sinclair, J., Hanks, P., et al.: The Collins Cobuild English Language Dictionary. HarperCollins, New York (1987)
6. The British National Corpus, version 3 (BNC XML Edition). Distributed by Oxford University Computing Services on Behalf of the BNC Consortium (2007)
7. Ruppenhofer, J., Baker, C.F., Fillmore, C.J.: The FrameNet Database and Software Tools. In: Braasch, A., Povlsen, C. (eds.) Proceedings of the Tenth Euralex International Congress, Copenhagen, Denmark, vol. I, pp. 371–375 (2002)
8. Palmer, M., Kingsbury, P., Gildea, D.: The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics 31(1), 71–106 (2005)
9. Herbst, T., et al.: A Valency Dictionary of English: a Corpus-Based Analysis of the Complementation Patterns of English Verbs, Nouns and Adjectives. De Gruyter, Berlin (2004)
10. Horák, A., Rambousek, A.: Server for Dictionary Editor and Browser (DEB) Platform (2008), http://deb.fi.muni.cz/
11. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch Engine. In: Proceedings of the Eleventh EURALEX International Congress, pp. 105–116. Universite de Bretagne-Sud, Lorient (2004)
12. Cinková, S., Toman, J., Hajič, J., Čermáková, K., Klimeš, V., Mladová, L., Šindlerová, J., Tomšů, K., Žabokrtský, Z.: Tectogrammatical Annotation of the Wall Street Journal. In: The Prague Bulletin of Mathematical Linguistics, vol. (92). Charles University in Prague (2009)
13. Rumshisky, A., Batiukova, O.: Polysemy in Verbs: Systematic Relations between Senses and their Effect on Annotation. In: COLING Workshop on Human Judgement in Computational Linguistics (HJCL 2008), Manchester (2008)

Extracting Human Spanish Nouns*

Sofia N. Galicia-Haro[1] and Alexander F. Gelbukh[2]

[1] Facultad de Ciencias, Universidad Nacional Autónoma de México, Mexico, D. F.
sngh@fciencias.unam.mx

[2] Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico, D. F.
gelbukh@gelbukh.com

Abstract. In this article we present a simple method to extract Spanish nouns with the linguistic property of "human" animacy. We describe a non-supervised method based on lexical patterns and on a person name list enlarged from a collection of newspaper texts. Results were obtained from the Web filters and estimation methods are proposed to validate them.

Keywords: Animacy, human mark, Spanish nouns, non supervised learning.

1 Introduction

In general, the animacy mark distinguishes living entities from non-living ones. But animacy might be considered as a range that goes from "human" consideration to "inanimate objects". For example, [17] analyze the codification of animacy in English. They distinct three categories: human beings, other animates and inanimates.

Animacy is an important category in linguistic analysis. Animacy has effects in grammar, in word order, in sentence production, etc. For example, in Spanish, reference to a direct object that is a human being makes the inclusion of the "a" preposition obligatory [1]; John Myhill discusses how in Chorti, a Mayan language that exhibits a strong tendency to VO order, animate subjects appear more in a preverbal position than inanimate subjects [16]; in English, control verbs in irregular past participles (e.g.: eaten, shaken) prefer animate subjects in active sentences [2].

For these reasons, in natural language processing, automatic animacy identification is important. Researchers have analyzed its importance in generation and translation [17], in parsing [12], in anaphora resolution [11], etc. Nevertheless, the animacy mark is not found systematically in common resources. Seeking in a Spanish dictionary such as DRAE[1] or MOLINER[2] the *coronel* 'colonel' noun, we find the same main description: *Jefe militar que manda un regimiento* 'Military chief that commands a regiment', but they do not mark explicitly its human condition, because it is clear for a human reader. Other nouns such as *basquetbolista* 'basketball player' (absent in DRAE) or *narcotraficante* 'drug dealer' (appearing as an adjective in DRAE), are nouns [+human]. So, it is not possible to extract this mark from such resources.

* Work done with partial support of Mexican Government (CONACyT, SNI, CGEPI-IPN).

[1] http://buscon.rae.es/draeI/

[2] http://www.diclib.com/cgi-bin/d1.cgi?l=es&base=moliner&page=showindex

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 75–83, 2010.

Researchers have conducted work to automatically identify animacy. For example, in [10], the authors present a method for English nouns using WordNet and machine learning techniques, and their evaluation results show that animate entities are more difficult to identify than inanimate. In [8], the authors develop a simple approach to discover gender and animacy knowledge. They automatically discover a large knowledge base of gender and animacy properties for noun phrases, with animacy based on a pronoun pattern for *who*.

The aim of our work is to annotate "human" animacy in Spanish texts. In this work, we obtain a list of "human" animacy nouns automatically. This paper is structured as follows. First we describe some characteristics of animacy in Spanish constructions. In Section 3 we present the proposed technique to obtain human animate nouns using instances like proper names. In Section 4 we present the results when applying this technique in Web-scale n-grams. Finally, in Section 5 we present the conclusions.

2 Animate Nouns in Spanish

Spanish, like other Indo-European languages, has grammatical gender: nouns are partitioned into sets which, in general, contrast distinctions of sex or animateness. Spanish has two genders (feminine, masculine), German has three genders (neuter, in addition). Identification of human animated nouns would be much easier if language could assign a precise gender for people. But languages like Spanish assign genders such as masculine and feminine to inanimate. Nouns denoting people, assigned to masculine or feminine gender according to sex[3], are a minority [6]. The "exceptions" (non-sexed objects assigned to either of those two genders) are the majority, thus making the semantic association a rather useless predictor for the gender of a noun.

Animate nouns in Spanish can be identified in different contexts which can be divided into several types:

- As a direct object

The animate direct object requires a preposition to be linked to the verb, becoming a prepositional complement. The so-called Prepositional Direct Complement is a topic much discussed in Spanish grammar. It is a linguistic phenomenon present in many languages [14], known as Differential Object Marking (DOM). In such languages, direct objects can be divided into two different classes and only one class receives a mark. In Spanish the mark corresponds to the preposition "a" when the noun is animate. For example:

Veo esa casa.	'I can see that house' (inanimate direct object)
Veo a esa niña.	'I can see that girl' (animate direct object)

This form is usual in standard Spanish. Nevertheless, in several dialects of Spanish, especially in Latin America, the preposition "a" precedes direct objects which are not animate if they are definitive and specific [15]. For example:

Vio a las sierras.	'He saw the saws' (Puerto Rican Spanish)
Cosecharon al maíz.	'They harvested the corn' (Argentinian Spanish)

[3] Author use the word "sex" to refer to biological gender, reserving "gender" for the grammatical category.

– According to the verb

Verbs select subjects according to their animate or inanimate condition. For example: *la madera cruje* 'the wood creaks', **Juan cruje* 'John creaks', although in some cases, the metaphorical sense could change a non-grammatical condition to a correct object selection. For example: ... *y crujo como sal que se derrite* '... and I rattle as salt that melts'.

In [3] the authors analyze the so-called verbs of temporal expression: *durar* 'to last' and *tardar* 'to take time'. Moliner's definition of the verb "to last" in its first meaning is "To be a thing happening, existing, working, etc., the time that expresses itself". Both are intransitive verbs and they may have a subject [+animate]. Thus, with respect to the verb *tardar*, in a phrase such as a *Julia tarda media hora en hacer un ejercicio* 'Julia takes half an hour doing an exercise', the verb makes reference to the time a subject [+ animate] employs in carrying out certain activity.

Nevertheless, such a syntactic structure (subject + verb + temporary complement + supplement) is impossible to realize with the verb "to last", which, in these cases, does not admit the feature [+animate]: the phrase is ungrammatical: **Julia dura treinta minutos en hacer un ejercicio* 'Julia lasts thirty minutes doing an exercise'. The authors indicate that, to be able to use the verb *durar* with a subject [+animate], it must be used in cases in which its meaning acquires other submeanings, with certain nuances that distinguish them from the first meaning: in these cases the verb *durar* means: to endure, to continue, to keep up.

In [14], the authors analyze different classes of verbs. They consider as their main sources of analysis the Bible and the Corpus del Español (s. XII to XIX). Their analysis confirms the hypothesis that the class of verb is the principal parameter for DOM in Spanish. Another important conclusion is that the direct object mark is determined by parameters in a multi-dimensional space.

– Pronouns

With pronouns, there is a tendency to use *le* like the pronoun of a direct object, although it is usually an indirect object pronoun or accusative pronoun, meaning 'to him/her', at the expense of the pronouns that are the accusative pronouns, when the referent is animate. DRAE[4] exposes that among other classes the so-called verbs of *psychic condition,* those that designate processes that affect encouragement or produce actions or emotive reactions, like affecting, scaring, amazing, convincing, etc., depending on different factors, admit the use of the accusative pronouns: *lo*(s), *la*(s), and the dative pronouns: *le*(s). The selection of one or other of these depends basically on whether the subject is or is not an active agent of the action and on the grade of volition that he has or assumes with regard to the action designated by the verb. With animate subjects this alternation can happen also if the action denoted by the verb is realized voluntarily or not for the subject. For example: *Su padre, que se había disfrazado, lo asustó* 'His father, who had disguised himself, scared him' (he gave him a fright intentionally), *Su padre, que se había disfrazado, le* asustó 'His father, who had disguised himself, scared him' (the fright is involuntary, the cause is the fact of going in disguise).

[4] http://buscon.rae.es/dpdI/SrvltGUIBusDPD?lema=le%EDsmo

- In noun apposition

An apposition is a construction of two close grammatical elements, the second of which specifies the first. In the juxtaposition of one noun to the other by means of the apposition, compounds are formed by two nouns which are written together (*compraventa* ‘buying and selling’) or separately (*compra venta*). In the case of animate nouns, a common apposition is that of a proper noun to another generic. This apposition specifies a personal characteristic (*lawyer Juan Torres, your brother Juan*). To create an information repository that helps to answer a question, [5] consider patterns to extract highly precise relationship information. The most productive patterns considered are two syntactic constructions that often indicate the relationship concept–instance, common noun-proper name and appositions, such as *President George Bush*.

3 Instances of Human Animate Nouns

The annotation of animacy is not standard in corpora or treebanks. Studies in the corpus on animacy, for example [10], have used data with manual annotations. Those annotations differ according to the considered scheme, with diverse granularity of categories. We consider in this work only one distinction between Human and Non Human, abbreviated as [+H] and [-H].

As [13] pointed out, the attributes of a given class can be derived by extracting and inspecting the attributes of individual instances from that class. For example, the attributes of the class Car are extracted by inspecting attributes extracted for Chevrolet Corvette, Toyota Prius, Volkswagen Passat, etc. Authors explain that this is particularly appealing when there are large sources of open-domain text (including the Web), since named entities are well represented on them and it is straightforward to obtain high-quality sets of instances automatically from such sources, among other reasons.

From Section 2 we can conclude that different parameters are required to formulate the rules that precisely determine animacy in Spanish nouns. So we propose to obtain the class of nouns [+H] identifying the contexts where human noun instances appear. For example, John is an instance of lawyers, of Pumas soccer players, of UNAM’s workers, etc. These classes (lawyers, players, workers) correspond to nouns with human mark.

3.1 Instances

Considering the work of Lin [9], where contexts are used to infer the meaning of an unknown word and are then employed to obtain similar words as an initial step in learning the definition of a word, we may find the contexts where proper names appear and from these obtain the animate nouns occupying the same contexts. For example, let us consider the following immersed phrases in sentences from Mexican newspaper texts:

1. . . . , *el éxito de* ***Francisco Céspedes*** *es indiscutible y* . . .
   ‘. . . , the success of Francisco Céspedes is indisputable and . . . ’
2. *El debate entre* ***Cuauhtémoc Cárdenas****y****Alfredo del Mazo*** *será cerrado* . . .
   ‘The debate between Cuauhtémoc Cárdenas and Alfredo del Mazo will be tough’

3. *Si el propio* ***Labastida*** *reconoce que hay tres equipos económicos que ...*
   'If Labastida himself admits that there are three economic teams that'
4. *..., pues fue expulsado por el árbitro* ***Eduardo Gasso*** *al acumular ...*
   '..., since he was expelled by the referee Eduardo Gasso on having accumulated'
5. *... golpe de Estado contra el entonces presidente* ***Carlos Andrés Pérez****, ...*
   '... coup d'état against the president of that time Carlos Andrés Pérez'

In these examples, the proper names in bold letters can be replaced by a general class. For example, Francisco Céspedes can be replaced by the *singer* animate noun, Cuauhtémoc Cárdenas and Alfredo del Mazo can both be replaced by *candidate*, Labastida by *economist*, etc. The last two examples correspond to appositions where the two nouns represent *class-instance*.

Since we first require a list of instances (simple personal names in opposition to name entities), we could select them from available lexicons, gazetteers or Web-derived lists of names. However, the proper name collection obtained in this way will be limited by the source used. To acquire a much wider list of proper names, we begin with a small list obtained from the Web, then increase this list using a two-step technique:

1. Extract animate nouns by means of patterns according to the linguistic rules for apposition and DOM that we describe in the following section.
2. The animate nouns obtained are used again in apposition and the obtained verbs in DOM patterns also are used to obtain new person names.

3.2 Patterns

We propose to extract simple proper names by means of patterns developed from the linguistic phenomena described in Section 2. From the four types described we decide to use patterns for direct object and noun apposition. Knowledge information in the syntactic and semantic levels of sentence analysis, in addition to full sentence context or even paragraph context, are required to determine nouns [+H] according to the verb and pronouns. Since we are interested in simpler methods to determine nouns [+H], we develop patterns where a narrow context is useful. For example:

VERB	"a"	PERSON_NAME	(*Veo a Juan* 'I see John)
DET	NOUN	PERSON_NAME	(*el poeta Rafael* 'the poet Rafael')

We apply these patterns to a text collection compiled from Mexican newspapers that are daily published in the WEB. The texts correspond to diverse sections: economy, politics, culture, sport, etc. from 1998 to 2002. The entire text collection has approximately 60 million words [7].

We wrote a program that applies such patterns ensuring that PERSON_NAME corresponds to a name from a list of Mexican person names with 456 elements obtained from the Web. The list has 191 masculine names (e.g.: *Aldo, Alejandro, Alfonso*), 178 feminine names (e.g.: *Abelina, Adela, Adelaida*) and 87 names of indigenous origin (e.g.: *Acamapichtli, Acatl, Acatzin*).

In the first step the following examples were obtained, among many others:
al administrador (Martín Ortega)
al doctor (Juan Ramón de la Fuente)

asesinaron a (José Francisco)
contestó a (Miguel López)

For the second step, the contexts like *al administrador, al doctor, asesinar a, contestar a*, etc., were used as patterns and the following names were extracted: *Alfonse, Alger, Álvaro, Amós*, among others. After applying the two-step technique and a manual revision of the results from newspaper texts we obtained a list of 836 person names that we call PERNAM, based on the contexts of 163 animate nouns.

4 Results: Set of Nouns [+H]

Using the instances of nouns [+H] it is possible to obtain their surrounding contexts and from the correct ones to automatically identify the nouns [+H] in texts. In many cases, the noun phrase in which the instance is included will be unambiguous and clearly associated with the semantic category. For example, soldier in a clear noun phrase context will always be a noun [+H]. In these cases, the noun phrase alone will be sufficient for the correct determination. In other cases, the context itself is not highly predictive and it will be ambiguous with regard to the semantic class.

First Step: Extraction of Person's Context

Google [4] released a collection of n-grams from Web pages. For a better analysis of animacy it would be necessary to examine the full context of every sentence. Nevertheless, in this work we use as corpora the Spanish 5-grams, considering that they are sufficient to capture the diverse structure of noun phrases and direct complements. Our work is as follows:

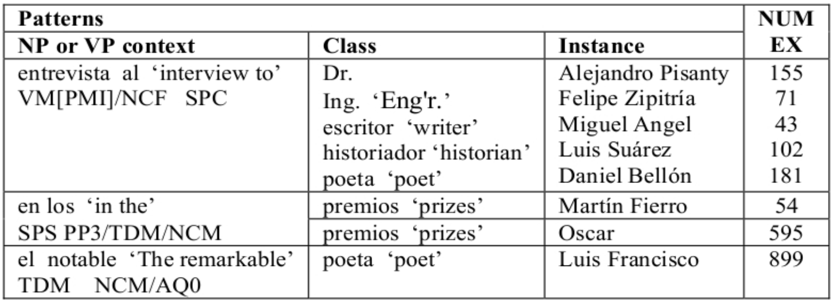
1. Extracting 5-grams having instances, that is, including person names verified in the PERNAM list
2. Assigning POS to each word without disambiguation and simple unknown words POS assignment.
3. Discarding those 5-grams with no clear cohesion between groups of words, that is, with conjunctions, punctuations, etc.
4. Sorting according to context similarities

An extract of the overall results is presented in Table 1. The first column shows the contexts with their POS, where: SP means preposition (SPC preposition abbreviation, SPS any other preposition), VM means verb (VMP participle, VMM imperative, VMI indicative), NC means common noun (NCF feminine, NCM masculine), PP3 is personal pronoun 3rd person, TDM means define article, AQ0 means qualifying adjective. The second column shows the possible class, i.e., noun [+H]. Column 3 gives the instance of the noun [+H] and column 4 shows the frequency of the 5-grams. All examples shown correspond to the apposition case, but other cases were found with direct object and pronoun *le*.

Second Step: Validation of Contexts

Our work is as follows:

Table 1. Some results obtained from the Spanish 5-grams



Patterns			NUM EX
NP or VP context	Class	Instance	
entrevista al 'interview to' VM[PMI]/NCF SPC	Dr.	Alejandro Pisanty	155
	Ing. 'Eng'r.'	Felipe Zipitría	71
	escritor 'writer'	Miguel Angel	43
	historiador 'historian'	Luis Suárez	102
	poeta 'poet'	Daniel Bellón	181
en los 'in the' SPS PP3/TDM/NCM	premios 'prizes'	Martín Fierro	54
	premios 'prizes'	Oscar	595
el notable 'The remarkable' TDM NCM/AQ0	poeta 'poet'	Luis Francisco	899

1. Each result was verified to make sure that the noun phrase context existed. For example, the following incorrect ones were discarded:
   <S> ahora como *Lola* , lit. 'now such as a Lola,'
   alquiler *apartamentos Santa Cruz de* lit. 'apartment rent Santa Cruz of'
   article thumbnail *Jesús no mira* lit. 'article thumbnail Jesus does not watch'
   mencionada *Norma Andina dispone :* lit. 'mentioned Norm Andean arranges:'
2. Each result consisting of a noun phrase was verified by means of concordance. For example, the following incorrect ones were discarded:
   de los reyes Alfonso IX 'of the kings Alfonso IX'
   the kings: masculine plural, Alfonso: masculine singular
   de malezas Carlos Gomez 03 'of undergrowths Carlos Gomez 03'
   malezas: feminine plural, Carlos: masculine singular
3. Each result consisting of a verb was verified as having a noun phrase before or after the verb.

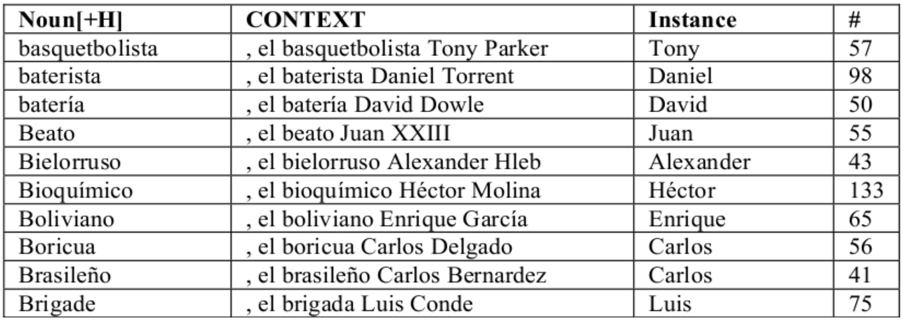
After this validation, contexts were used to find possible nouns [+H] and, as the authors in [5] indicated, the most productive pattern was that of noun apposition.

Third Step: Validation of Nouns[+H]

The derived class may include a lot of noise. For example: *premios* 'prizes' in column 2 of Table 2 corresponds to a noun [-H], and thus filters or estimation methods are required for knowledge discovery. We propose a very simple method: searching in the Web for the opposite linguistic pattern with a common instance:

1. Noun phrase context:
   For noun validation, a Web search for the verb pattern (VERB "a" NOUN [+H]) was launched. For example, the following incorrect ones were discarded:
   de metro Pedro de Valdivia launches"vio a metro" with 1 hit
   lado íntimo Cecilia Bolocco Angelina launches "vio a lado íntimo" with 0 hits
2. Verb context
   For noun validation in a verb context, a Web search for the apposition pattern (DET NOUN[+H] *Juan*) was launched. For example:
   anunció el ingeniero Miguel launches "el ingeniero Juan" with 1,330,000 hits
   llega cedido David Lizoain . launches "el cedido Juan" with 3 hits
   The threshold for accepting noun [+H] was set at 50 snippets.

Table 2. An extract of the list of Nouns[+H]



Noun[+H]	CONTEXT	Instance	#
basquetbolista	, el basquetbolista Tony Parker	Tony	57
baterista	, el baterista Daniel Torrent	Daniel	98
batería	, el batería David Dowle	David	50
Beato	, el beato Juan XXIII	Juan	55
Bielorruso	, el bielorruso Alexander Hleb	Alexander	43
Bioquímico	, el bioquímico Héctor Molina	Héctor	133
Boliviano	, el boliviano Enrique García	Enrique	65
Boricua	, el boricua Carlos Delgado	Carlos	56
Brasileño	, el brasileño Carlos Bernardez	Carlos	41
Brigade	, el brigada Luis Conde	Luis	75

List of Nouns [+H]

In Table 2 some portions of the resulting list of nouns [+H] are presented. We can observe that some nouns neglected in traditional dictionaries are present in our results, for example: *basquetbolista, clavadista, perredista*, etc. The total results obtained are 57,808 noun [+H] contexts.

We made a small manual evaluation. We collected three Mexican newspaper articles corresponding to 22/12/04. The texts contain 1,154 words and 74 nouns [+H]. After assigning POS to each word without disambiguation and simple unknown words POS assignment we applied our described results: the list of nouns [+H] and contexts. We obtained 0.77 precision and 0.81 recall, where:

Precision: # of correct noun [+H] detected / # of noun [+H] detected

Recall: # of correct noun [+H] detected / # of noun [+H] manually labeled

We found that among the four nouns bad recognized two cases correspond to nouns appearing in the nouns [+H] list but within a non person context. For example: *con el grado de capitán de Ejército* 'with the degree of captain of Army'. The other two cases correspond to bad proper name detection. For all the non-detected nouns [+H] the context does not help and the nouns do not appear in the list.

5 Conclusions

The above linguistic descriptions have shown that animacy for Spanish as in other languages depends on diverse features. The chosen techniques based on morphosyntactic features of animacy have proven to extract the human class well. As we have seen, the specification in nouns and the direct object provide stable structures for animacy even in narrow contexts such as those of the 5-grams. Two experiments have been described above which indicate that instances can be used to capture generalizations which pertain to nouns [+H].

We must emphasize that very frequent nouns [+H] are usually well described in other lexical resources but many others are not described in detail or even neglected. The contributions of this paper are: the attempt to discover animacy noun knowledge from very narrow contexts for Spanish nouns and detecting verbs with an animate subject or animate direct object; they are based on unsupervised methods.

References

1. Aissen, J.: Differential Object Marking: Iconicity vs. Economy. Natural Language and Linguistic Theory 21(3), 435–483 (2003)
2. Altmann, L.J.P., Kemper, S.: Effects of Age, Animacy, and Activation Order on Sentence Production. Language and Cognitive Processes 21(1), 322–354 (2006)
3. Berenguer, C.R., Cruz Pastor Ferrán, M.: ¿Cuánto dura/tarda la clase de Español?: una reflexión sobre determinados usos verbales en Español. In: Lengua y cultura en la enseñanza del Español a extranjeros. Actas del VII Congreso de ASELE, pp. 397i–406i. Ediciones de la Universidad de Castilla la Mancha (1998)
4. Brants, T., Franz, A.: Web 1T 5-gram Version 1 Linguistic Data Consortium (2006)
5. Fleischman, M., Echihabi, A., Hovy, E.: Offline Strategies for Online Question Answering: Answering Questions before They are Asked. In: Proceedings of the ACL Conference, pp. 1–7 (2003)
6. Foundalis, H.E.: Evolution of Gender in Indo-European Languages. In: Proceedings of the 24th Annual Conference of the Cognitive Science Society, Fairfax, VA, pp. 304–309 (2002)
7. Galicia-Haro, S.N.: Using Electronic Texts for an Annotated Corpus Building. In: 4th Mexican International Conference on Computer Science, ENC, Mexico, pp. 26–33 (2003)
8. Heng, J., Lin, D.: Gender and Animacy Knowledge Discovery from Web-Scale N-Grams for Unsupervised Person Mention Detection. In: Proceedings of PACLIC (2009)
9. Lin, D.: Automatic Retrieval and Clustering of Similar Words. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 768–774 (1998)
10. Orăsan, C., Evans, R.: Learning to Identify Animate References. In: Proceedings of the Workshop on Computational Natural Language Learning, ACL (2001)
11. Orăsan, C., Evans, R.: NP Animacy Resolution for Anaphora Resolution. Journal of Artificial Intelligence Research 29, 79–103 (2007)
12. Øvrelid, L.: Empirical Evaluations of Animacy Annotation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pp. 630–638 (2009)
13. Paşca, M., Van Durme, B.: What You Seek Is What You Get: Extraction of Class Attributes from Query Logs. In: Proceedings of the International Joint Conference on Artificial Intelligence 2007, pp. 2832–2837 (2007)
14. von Heusinger, K., Kaiser, G.A.: Differential Object Marking and the Lexical Semantics of Verbs in Spanish. In: Kaiser, G.A., Leonetti, M. (eds.) Proceedings of the Workshop Definiteness, Specificity and Animacy in Ibero-Romance Languages, pp. 85–110 (2007)
15. von Heusinger, K., Kaiser, G.A.: The Interaction of Animacy, Definiteness and Specificity in Spanish. In: von Heusinger, K., Kaiser, G.A. (eds.) Proceedings of the Workshop: Semantic and Syntactic Aspects of Specificity, Romance Languages, pp. 41–65. Universität Konstanz, Konstanz (2003)
16. Yamamoto, M.: Animacy and Reference: A Cognitive Approach to Corpus Linguistics. Studies in Language Companion Series, vol. 46. John Benjamins, Amsterdam (1999)
17. Zaenen, A., Carletta, J., Garretson, G., Bresnan, J., Koontz-Garboden, A., Nikitina, T., O'Connor, M.C., Wasow, T.: Animacy Encoding in English: Why and How. In: Proceedings of the 2004 ACL Workshop on Discourse Annotation, pp. 118–125 (2004)

Semantic Duplicate Identification with Parsing and Machine Learning

Sven Hartrumpf[1], Tim vor der Brück[1], and Christian Eichhorn[2]

[1] Intelligent Information and Communication Systems (IICS)
FernUniversität in Hagen, 58084 Hagen, Germany
{sven.hartrumpf,tim.vorderbrueck}@fernuni-hagen.de
[2] Lehrstuhl Informatik 1, Technische Universität Dortmund,
44227 Dortmund, Germany
christian.eichhorn@tu-dortmund.de

Abstract. Identifying duplicate texts is important in many areas like plagiarism detection, information retrieval, text summarization, and question answering. Current approaches are mostly surface-oriented (or use only shallow syntactic representations) and see each text only as a token list. In this work however, we describe a deep, semantically oriented method based on semantic networks which are derived by a syntactico-semantic parser. Semantically identical or similar semantic networks for each sentence of a given base text are efficiently retrieved by using a specialized index. In order to detect many kinds of paraphrases the semantic networks of a candidate text are varied by applying inferences: lexico-semantic relations, relation axioms, and meaning postulates. Important phenomena occurring in difficult duplicates are discussed. The deep approach profits from background knowledge, whose acquisition from corpora is explained briefly. The deep duplicate recognizer is combined with two shallow duplicate recognizers in order to guarantee a high recall for texts which are not fully parsable. The evaluation shows that the combined approach preserves recall and increases precision considerably in comparison to traditional shallow methods.

1 Introduction

With the growth of the web, the number of available texts has increased rapidly.[1] The number of duplicates increased with similar speed, deliberately by generating plagiarisms or unwittingly by presenting information already given by other users or services.

To detect duplicates is a relevant task for many different areas: applications regarding information access like search engines or question answering systems try not to response with duplicate information to user requests. Copyright owners and tutors want to find cases of copyright violations and plagiarism even if the violator used techniques to obfuscate the source. Other uses could include backup tools trying to

[1] We would like to thank all members of our departments who supported us. This work was funded by the DFG project *Semantische Duplikatserkennung mithilfe von Textual Entailment* (HE 2847/11-1).

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 84–92, 2010.

eschew redundant files or computer administrators searching for redundant files which can be deleted in order to save disk space.

In prior work, duplicate detection employs shallow methods, working on surface-oriented factors or features only, which are mainly derived from n-grams, rare words and spelling errors, with n-grams being used most frequently [1, Sect. 3.2]. Even if the capability of these approaches has increased, they are still capable of detecting only three quarters of the tested plagiarisms [2].

Using the semantics of words, sentences, paragraphs, or even whole texts, two texts which are semantic duplicates, i.e., expressing the same content without sharing many words or word sequences and hence without having similar values of shallow features, can be tackled. Since shallow checkers can easily be tricked by experienced users which employ advanced paraphrase techniques, a deep approach that compares full semantic representations[2] of two given texts is designed, implemented, and evaluated in the SemDupl (Semantic Duplicate) project in order to detect even obfuscated plagiarisms and semantic duplicates.

2 State of the Art

As stated, detecting duplicates is of high interest for holders of rights and tutors. Therefore, many tools exist to detect plagiarisms in given corpora or the web. Below are some of the best ranked systems according to the 2008 test of the University of Applied Sciences Berlin (FHTW) [4].

Copyscape[3], a plagiarism checker of Indigo Stream Technologies Ltd. Given a text it searches the Internet for possible plagiarism of this text using the document's words in the given order.

Plagiarism Detector[4] by SkyLine, Inc. uses non-overlapping n-grams with a configurable spacing between them to find online-plagiarisms of a given text in various possible input formats.

Urkund[5] by PrioInfo AB targets to check papers written by students for possible plagiarism and searches the Internet (with known paper mills), an own corpus of scientific publications and papers checked for plagiarism before.

WCopyfind[6] is an n-gram based plagiarism checker of the University of Virginia, Charlottesville [5]. It targets student's papers, searching a corpus which has to be compiled by the user. Since it is open source software this tool was used as a comparison for our SemDupl system.

3 The SemDupl Corpus

The corpus used in the learning process of the shallow duplicate checker CErkenner (see Sect. 4.1) and for evaluation purposes (see Sect. 8) contains 287,044 words in 13,622 sentences. It is composed of the following manually annotated subcorpora:

[2] The formalism is MultiNet, Multilayered Extended Semantic Networks, [3].

[3] `http://www.copyscape.com/`, first and third place (premium and free version).

[4] `http://plagiarism-detector.com/`, scored second place.

[5] `http://www.urkund.de/`, scored fourth place.

[6] `http://plagiarism.phys.virginia.edu/Wsoftware.html`, marked as "good".

RSS news (semdupl-rss) News feed articles of different German media consisting of 99 texts annotated with 113 duplicate pairs[7].

Prose (semdupl-prose) Short stories by Edgar Allen Poe translated to German by different translators (split in 136 parts of about 600 words each) with 68 duplicate pairs.

Internet (semdupl-google) 100 texts collected from Google (the 10 top texts of the 10 fastest growing search terms in 2008), containing 42 duplicate pairs.

Plagiarism (fhtw) Weber-Wulff's collection of plagiarisms (slightly extended), annotated as 77 texts with 39 duplicate pairs.

4 Shallow Approaches

SemDupl uses two shallow approaches as filters on large corpora and/or as a robust fall-back strategy (if deep parsing fails).

4.1 CErkenner (CE)

To detect whether a text is the duplicate of another text, CE uses set of 39 features derived from the surface structure of the texts which include the following:

Word sets The words of the compared texts represented as sets, with elements being the text's words as they are given, without stop words, in stemmed form or united with their synonyms.

Typos Weber-Wulff [6] shows that a promising strategy in searching for plagiarism is to compare the spelling mistakes in different texts, since if a text is plagiarized, its typos are often copied, too.

Length of words and sentences Weber-Wulff [6] states that plagiarized texts often share the same style of writing. Since a writers style includes the average length of words and sentences (per paragraph), these two values are used as features in the process.

N-grams Word n-grams are sequences of *n* words from the texts. In CErkenner the different types of word n-grams used are simple n-grams, alliterations (n-grams where all words start with the same letter), phonetic alliterations (alliterations with the words sharing the same initial phoneme) and k-skip-n-grams (n-grams where up to *k* words are left out ("skipped") between the elements of the n-grams [7]).

CE combines these features using machine-learning techniques and was trained using the SemDupl corpus (see Sect. 3).

4.2 ShallowChecker (SC)

Tests indicated that the shallow approach of CE achieves good results regarding precision and accuracy, but due to its time complexity it is rather unsuited for large

[7] The reported numbers include only non-trivial duplicates, i.e., the pairs made of the same document and symmetric variations are excluded.

corpora. So another shallow approach was devised using only features which can be calculated efficiently.

In the preprocessing phase, the ShallowChecker (SC) searches the given texts for misspelled words and words with a frequency class above a given threshold and compiles all n-grams with lengths from 3 to 7. These values are used as indices whereas the text's id (e.g. filename) is used as value. This generates a database with a list of texts for each value (with a table for each feature).

In the detection phase all rows r containing a given text are searched inside the tables. For each *other* affected text found inside the rows the ratio between the total number of rows r and the number of rows in r containing the affected text is calculated for each table (and therefore feature). These scores are combined linearly and normalized, resulting in an combined score for each text pair. A text is regarded as a duplicate if the score is greater than a given threshold.

4.3 Comparison of Shallow Approaches

CErkenner works on texts without any major preprocessing: it is ready to instantly check an arbitrary pair of texts without *any* preprocessing steps as an "out-of-the-box" duplicate detector. Its capability to learn the "definition" of duplicates on an annotated corpus leads to a detection which has a lower chance of failing because of bad user-set thresholds. Its downside is it has to inspect every possible text pair in order to detect all duplicates in a given corpus, resulting in quadratic time complexity, so it should be used on small corpora. ShallowChecker, on the other hand, uses preprocessing resulting in a lower time complexity while detecting, but only some of the possible features can be used as index values and the thresholds, which are defined by the user, may, if not set well, become a source of errors.

5 Linguistic Phenomena Relevant for Semantic Duplicates

5.1 Types of Paraphrases for Semantic Duplicates

Many problems exist for standard surface oriented comparisons for duplicate detection; here are some examples:

1. different word forms due to inflection
2. different orthography (e.g. new and old orthography in German).
3. abbreviations/acronyms and expanded forms
4. different hyphenation of compounds
5. different word order (especially relevant in German)
6. discontinuous word forms (e.g. German verbs with separable prefix)
7. different voices (active or passive in German)
8. nominalization of situations, e.g. *discussion* vs. *to discuss*; supported by around 3,000 verb-noun links in the lexicon (HaGenLex, [8])
9. information distribution across sentences
10. synonyms: partially solved by HaGenLex plus GermaNet (relation SYNO)
11. hyponyms: e.g. *dentist* vs. *physician* solved by lexico-semantic relations.

12. compounds vs. analytical expressions like complex NPs and clauses: e.g. *finance gap* vs. *gap in financing* (ca. 500,000 compound analyses available)
13. idioms: An idiom lexicon of 250 idioms based on verbs is employed.
14. support verb constructions (SVCs), e.g. *to utter an objection* vs. *to object*. In SemDupl, this achieved by around 500 MultiNet rules (derived from a SVC lexicon) that are applied during query expansion.
15. coreferences (different expressions referring to the same entity): solved by the coreference module.
16. entailments e.g. *to buy* vs. *to sell*; covered in part by entailments from HaGenLex and entailments derived from knowledge bases like GermaNet and manual translations of XWordNet (several thousand rules).

Most of the above paraphrase problems are tackled by the WOCADI parser (see Sect. 7) and its modules; limitations have been mentioned above.

A nice example from our semdupl-prose subcorpus shows that these phenomena combine quite often: *... sagte Dupin, während er seinem Besuch eine Pfeife reichte und einen bequemen Sessel hinschob./ Dupin ... said, while he passed his visitor a pipe and moved a comfortable chair to him* vs. *... antwortete Dupin, während er den Gast mit einer Pfeife versorgte und einen bequemen Sessel heranschob. Dupin ... replied, while he provided his guest with a pipe and moved a comfortable chair up to him* vs. The two sentences can only be reliably linked as nearly synonymous if four links can be constructed:

1. *hinschieben* and *heranschieben* can be linked as cohyponyms;
2. *reichen/to pass* can be related to *versorgen/to provide* via verb entailment represented at *versorgen* and a troponym for *reichen*;
3. *antworten/to reply* as a troponym of *sagen/to say*; and
4. *Gast/guest* and *Besuch/visit(or)* as synonyms.

5.2 Restrictive Contexts and Other Precision Problems for Semantic Duplicates

Precision is less of a problem for a deep approach; nevertheless some phenomena must be controlled to preserve precision even in a deep approach:

1. incorrect phrases: solved by parsing sentences
2. incorrectly selected reading (wrong reading of ambiguous word or constituent)
3. negation; constituent negation (compatibility test for the FACT layer feature in MultiNet suffices); sentence negation, similarly.
4. other modalities. Incompatible modalities are tested in the semantic networks. Similarly, hypothetical situations must be excluded from matching real situations. Other examples of modality come from epistemic modals like *glauben/to believe*.

6 Knowledge Acquisition for Deep Duplicate Detectors

The deep duplicate detector can only be as good as the underlying knowledge bases. Therefore, the SemDupl project tries to (1) consolidate our existing knowledge sources, (2) automatically (or semi-automatically) derive new knowledge bases, and (3) validate these new knowledges bases.

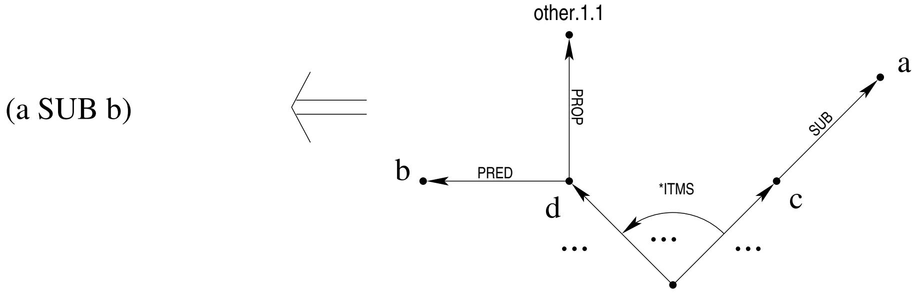


Fig. 1. Deep pattern for hypernymy extraction (premise as a semantic network)

6.1 Hypernym Acquisition

A type of near-duplicates that is both quite easy to create and to detect is a pair of sentences being almost identical except that certain words (or concepts on a semantic level) of the original sentence are replaced by hypernyms. This is a method often used while trying to obfuscate plagiarism. For example, *His father buys a new laptop.* implies *His father buys a new computer.* In the second sentence, *laptop* is replaced by one of its hypernyms, *computer*. Thus, a large collection of hypernyms is quite vital for near-duplicate recognition.

Since Wikipedia is often used as source for plagiarisms or duplicates, hypernyms and holonyms are extracted from Wikipedia using a pattern-based approach, differentiating between shallow and deep patterns.

Both types of patterns consist of a conclusion part of the form (a SUB b) which specifies that, if the premise holds, a hypernymy relationship between the concepts which are assigned to the variables a and b holds. The assignments for both variables are determined by matching the premise part to a linguistic structure which is created by analyzing the associated sentence.

The premise of a shallow pattern is given just by a regular expression which is tried to be matched with the token list of a sentence. In contrast, the premise of a deep pattern is given as a semantic network graph. This graph is tried to be matched to the semantic network of a sentence by a graph pattern matcher (or an automated theorem prover if axioms are to be employed). An example pattern is given in Equation (1) and Fig. 1.

$$(a \text{ SUB } b) \leftarrow \underset{*\text{ITMS}}{follows}(c,d) \wedge (d \text{ PRED } b) \wedge (d \text{ PROP} \mathit{other.1.1}) \wedge (c \text{ SUB } a) \qquad (1)$$

$follows_{*\text{ITMS}}(c,d)$ denotes the fact that c precedes d in the argument list of the function *ITMS. This pattern can be employed to extract the hypernymy relation (*cello.1.1* SUB *instrument.1.1*) from the sentence: *The old man owns a cello and other instruments*. Note that we consider *instance of* relations as a special kind of hypernymy as well and such relations were also extracted by our algorithm.

6.2 Deep vs. Shallow Patterns

On the one hand, a shallow pattern has the advantage that it is also applicable if the parse fails. It only relies on the fact that the tokenization is successful. On the other hand, deep patterns are still applicable if there are additional constituents between hyponym and hypernym, where shallow patterns often fail.

Another advantage of deep patterns is illustrated by the following sentence: *In any case, not all incidents from the Bermuda Triangle or from other world areas are fully explained.* From this sentence, a hypernymy pair cannot be extracted by the Hearst pattern *X or other Y* [9]. The application of this pattern fails due to the word *aus/from* which cannot be matched. To extract this relation by means of shallow patterns an additional pattern would have to be introduced. This could also be the case if syntactic patterns were used instead since the coordination of *Bermuda Triangle* and *world areas* is not represented in the syntactic constituency tree but only on a semantic level[8]. Thus, the same deep pattern can be used for the hypernymy extraction in this sentence as for the following phrase: *the Bermuda Triangle or other world areas*.

Furthermore, different syntactic or surface representations are frequently mapped to the same semantic network, e.g.:

1. *He owns a cello, a violin and other instruments.*
2. *He owns a violin, a cello as well as other instruments.*

Thus, the hyponymy relationships that cellos and violins are instruments can be extracted by the application of the same deep pattern. However, to extract the same information by the application of shallow patterns two different patterns have to be defined. Finally, the deep approach allows the usage of logical axioms, which can make, by using inferences, the patterns more generally applicable.

7 Deep Duplicate Detector (DC)

To handle linguistic phenomena adequately, i.e., identify paraphrase phenomena discussed in Sect. 5.1 and to not get disturbed by non-paraphrase phenomena discussed in Sect. 5.2, a deep semantic approach to duplicate detection has been developed. It integrates existing tools for producing semantic representations for texts: the WOCADI parser and the CORUDIS coreference resolver [10]. In an indexing phase, all texts in the base corpus are transformed into semantic representations by WOCADI and CORUDIS.

In the detection step, the duplicate candidate (text) is analyzed in the same way as the texts of the base corpus. For each sentence in the candidate, a semantic search query is sent to a retrieval system that contains all the semantic representations for the base corpus. Matches are collected and after all sentences of the candidate have been investigated, scores are calculated from the results for the text sentences. The average overlap score over all candidate sentences is a good score. The individual overlap score is calculated by the retrieval system, based on distances of related concepts and the distance between the left-hand side and right-hand side of inference rules. The average detection time (for parsing of the candidate text and querying the SemDupl corpus) was around 30 seconds on a PC with one CPU core.

[8] Note that some dependency parsers normalize some syntactic variations too.

Table 1. Confusion matrices for shallow and deep approaches. D=Duplicate, ND=No Duplicate.

	SC		CE		SC+CE	
	D	ND	D	ND	D	ND
D	97	157	200	54	201	53
ND	16	21,637	14	21,639	13	21,640

	DC		DC+SC		DC+SC+CE	
	D	ND	D	ND	D	ND
D	42	212	106	148	202	52
ND	5	21,648	16	21,637	11	21,642

Table 2. F-measure, precision, recall, and accuracy

	Shallow approaches				Deep approaches		
Measure	WCopyFind	SC	CE	SC+CE	DC	DC+SC	DC+SC+CE
F-measure	0.259	0.529	0.855	0.859	0.279	0.564	0.865
Precision	0.830	0.858	0.935	0.939	0.894	0.869	0.948
Recall	0.154	0.382	0.787	0.791	0.165	0.417	0.795
Accuracy	0.990	0.992	0.997	0.997	0.990	0.993	0.997

8 Evaluation

The three individual detectors as well as the combined system have been evaluated on the SemDupl corpus (see Sect. 3), which is annotated for duplicates. For each text pair and each approach, a set of features values is generated where high values indicate the texts being duplicates. These values are combined by the support vector machine classifier WLSVM [11], which is based on libsvm [12]. For training this classifier, the text pairs of our corpus were used (in ten-fold cross-validation). The confusion matrices calculated for shallow and deep approaches are shown in Table 1.

9 Interpretation and Conclusion

In order to compare the results of the combined system with plagiarism detection software *WCopyFind* was evaluated on our text corpus, too. Table 2 shows the results of our approaches. Precision is the relative frequency that a system-reported duplicate pair is a duplicate pair in the gold standard annotation. In contrast, accuracy looks at all decisions for given document pairs: the relative frequency that a document pair is correctly classified as duplicate or non-duplicate. Accuracy values are very high because the class of duplicates is tiny compared to the class of non-duplicates.

Except for the single DC approach, each approach of our system generates significantly better results in terms of precision, recall, and F-measure than *WCopyFind* (significance level of 1%). Note that the DC approach can show its full potential only in more professionally constructed duplicates. Furthermore, F-measure, precision, and recall of the combined shallow+deep system are significantly higher (significance level of 5%) than for the combination of the two shallow systems. We want to improve the coverage of the deep approach by further extending its knowledge bases by (semi-) automatic means.

References

1. Eichhorn, C.: Automatische Duplikatserkennung (Automatic Duplicate Detection). Der Andere Verlag, Tönning (2010)
2. Hüttinger, G.: Software zur Plagiatserkennung im Test – die Systeme haben sich deutlich gebessert (Test of plagiarism detection software) (November 2008), http://www.htw-berlin.de/Aktuelles/Pressemitteilungen/2008/index.html
3. Helbig, H.: Knowledge Representation and the Semantics of Natural Language. Springer, Berlin (2006)
4. Weber-Wulff, D.: Softwaretest 2008 (2009), http://plagiat.fhtw-berlin.de/software
5. Balaguer, E.V.: Putting Ourselves in SME's Shoes: Automatic Detection of Plagiarism by the Tool. In: Proceedings of PAN Workshop and Competition, Valencia, Spain (2009)
6. Weber-Wulff, D.: Der große Online-Schwindel (The Big Online Deception). Spiegel Online (2002)
7. Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y.: A Closer Look at Skip-Gram Modelling. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC), Geneva, Switzerland, pp. 1222–1225 (2006)
8. Hartrumpf, S., Helbig, H., Osswald, R.: The Semantically Based Computer Lexicon HaGenLex – Structure and Technological Environment. Traitement automatique des langues 44(2), 81–105 (2003)
9. Hearst, M.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of COLING, Nantes, France (1992)
10. Hartrumpf, S.: Hybrid Disambiguation in Natural Language Analysis. Der Andere Verlag, Osnabrück (2003)
11. EL-Manzalawy, Y., Honavar, V.: WLSVM: Integrating LibSVM into Weka Environment (2005), Software available at http://www.cs.iastate.edu/~yasser/wlsvm
12. Chang, C.C., Lin, C.J.: LIBSVM: a Library for Support Vector Machines (2001), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Comparison of Different Lemmatization Approaches through the Means of Information Retrieval Performance*

Jakub Kanis and Lucie Skorkovská

Univ. of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics
Univerzitní 8, 306 14 Pilsen, Czech Republic
{jkanis,lskorkov}@kky.zcu.cz

Abstract. This paper presents a quantitative performance analysis of two different approaches to the lemmatization of the Czech text data. The first one is based on manually prepared dictionary of lemmas and set of derivation rules while the second one is based on automatic inference of the dictionary and the rules from training data. The comparison is done by evaluating the mean Generalized Average Precision (mGAP) measure of the lemmatized documents and search queries in the set of information retrieval (IR) experiments. Such method is suitable for efficient and rather reliable comparison of the lemmatization performance since a correct lemmatization has proven to be crucial for IR effectiveness in highly inflected languages. Moreover, the proposed indirect comparison of the lemmatizers circumvents the need for manually lemmatized test data which are hard to obtain and also face the problem of incompatible sets of lemmas across different systems.

1 Introduction

The task of automatic lemmatization, i.e. finding the “lexical headword” of a given word form, is one of the tasks that are especially important for the highly inflected languages such as Czech where the abundance of word forms pertaining to a single lemma complicates many of natural language processing tasks, ranging from the language modeling (where it causes unfavorable fragmentation of the training data) to the tasks of keyword spotting and information retrieval (IR), where the (very frequent) mismatch between the word form used in the query and the word forms occurring in the searched collection prevents many keyword occurrences and/or relevant documents from being found. The importance of the lemmatizers for IR effectiveness that was revealed by the previous experiments together with the fact that the intrinsic evaluation of the lemmatizers (i.e. measuring their performance on the manually annotated gold standard data) faces the issues of possibly incompatible lemma sets in various systems prompted us to try to evaluate the performance of different lemmatizers extrinsically – by

* This research was supported by the Min. of Education of the Czech Republic, project. No. MŠMT LC536, by the grant of the University of West Bohemia, project No. SGS-2010-054 and by the Grant Agency of Academy of Sciences of the Czech Republic., project. No. 1ET101470416.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 93–100, 2010.

measuring their effect on the results of another task, in this case the information retrieval. Furthermore, we also wanted to test our hypothesis that the quality of our automatically trained lemmatizer (measured through the means of IR performance) is fully comparable with the quality of the lemmatizer employing carefully prepared handcrafted dictionary, even thought the intrinsic performance measures suggest the superiority of the latter system. If such hypothesis is corroborated, it would hint that the researchers who would be developing lemmatizers for IR purposes in new languages do not have to implement a perfect handcrafted lemmatizer but could rely on the automatically trained one whose development is much faster.

2 Description of Lemmatizers

There are two main processes used for derivation of new words in a language: the inflectional and the derivative process. The words are derived from the same morphological class (for example the form *cleared* and *clears* of the verb *clear*) in the inflectional process while in the derivative process are derived from other morphological classes (*clearly*). The creation of a new word can be reached by applying a set of derivation rules in the both processes. The rules provide adding or stripping prefixes (prefix rule) and suffixes (suffix rule) to derive a new word form. From this point of view, the lemmatization can be regarded as the inverse operation to the inflectional and derivative processes.

We will compare two different approaches (manual versus automatic) to the lemmatizer construction and its influence on IR system in our experiments. For this purpose we use two different lemmatizers. The first one is based on the handcrafted dictionary of lemmas and set of affix (prefix and suffix) patterns. The second one is automatically trained lemmatizer.

2.1 Handcrafted Lemmatizer

The state-of-the-art Czech morphological analyzer which is available as part of The Prague Dependency Treebank[1] [1] was selected as a representative of handcrafted lemmatizers. The analyzer provides all possible lemmas for a given word and also a set of all conceivable morphological tags. The analyzer uses the handcrafted dictionary of lemmas (228,000 [2]) and manually created set of affix patterns (As is the author's best knowledge).

2.2 Automatically Trained Lemmatizer

Automatically created lemmatizer employed in our experiments is a slightly modified version of the lemmatizer introduced in [3]. This lemmatizer uses a dictionary of lemmas and a set of affix rules, both automatically inferred from training data. The training data consist of (full word form – lemma) pairs. The inference of lemmatization rules is based on searching for the longest common substring of the full form and the

[1] We used the tool from version 2.0 of the treebank concretely.

lemma. The lemmatization rules are in the form of if-then rules (for example, a simple lemmatization rule is: if a word ends by E, then strip E and add ION, i.e. in the symbolic form: $E > -E, ION$).

The main modification of the lemmatizer involves adding new patterns for lemmatization of out-of-vocabulary (OOV) words, that is, the word forms that were not seen in the training dictionary. There are actually two types of OOVs — the ones whose lemma is missing in the training data as well and the ones whose lemma occurs in the training set but not in pair with the word form in question. The new patterns for OOV words arise by concatenation of particular prefix and suffix pattern for each pair (full word form - lemma) in the training data. In the previous version of the lemmatizer, the prefix and suffix patterns were used separately. So now the lemmatization of unknown words involves the creation of prefix and suffix patterns (the chain of applicable prefix or suffix rules) and its concatenation. Then the concatenated pattern is firstly searched in the pattern library and if it does not exist then particular prefix and suffix patterns are searched in the library. If the particular pattern is found in the library then a most probable rule associated with this patterns is used to process the given word.

2.3 Training Data and Comparison of Lemmatizers

Data from two different sources were used for training of the automatically constructed lemmatizer. The first source was the previously mentioned Prague Dependency Treebank 2.0 (PDT). The second one was the Czech dictionary of lemmas and derivation rule file from the spell-checking program Ispell [4]. PDT contains full word forms and the corresponding lemmas, thus the training data were obtained by simply extracting these pairs. The second set of training data was prepared from Ispell files by using our own morphological generator (the Ispell files contain rules that allow to generate all full word forms for each lemma in the dictionary). In Tab. 1 are the quantitative informations about both acquired training data sets (PDT and Ispell) and in Tab. 2 are the informations about automatically created lemmatizers (Lem_PDT and Lem_Ispell).

Comparison of the accuracy of different lemmatizers is a difficult task due to the need for the manually lemmatized test data. In addition, the evaluation of results should be done manually as well because different lemmatizers generally do not share the same set of lemmas. Strictly speaking, the lemma is usually the infinitive for the verbs and the word in masculine, singular and nominative form for other inflected part-of-speech types, but generally each word form can be chosen as lemma for the group of words with the same stem. This selection heavily depends on the decision made by the dictionary author or the training data annotator. We have proposed an indirect comparison of the lemmatizers through set of IR experiments for these reasons which will be described in the next section.

The direct comparison is of course possible (and often performed) when the lemmatizers do share the same set of lemmas. Since this is the case of the handcrafted lemmatizer (Lem_H) and the lemmatizer trained on PDT training data (Lem_PDT), we have compared them directly and the results are in Tab. 3. Recall (the number of the correctly lemmatized words to the number of all processed words ratio) (R), precision (the ratio of the number of the correctly lemmatized words to the number of all lemmas generated by the lemmatizer for all correctly lemmatized words) (P) and a

Table 1. PDT and Ispell training data

Training set	# pairs	# lemmas
PDT	200,431	66,401
Ispell	4,315,161	297,701

Table 2. Automatically created lemmatizers

	Lem_PDT	Lem_Ispell
# lemmas	66,401	297,701
# rules	2,431	2,683
# P rules	213	55
# S rules	2,218	2,628
# patterns	28,867	34,999
# P+S patterns	26,436	32,331
# P patterns	213	55
# S patterns	2,218	2,613

harmonic F-measure $((2 \cdot R \cdot P)/(R+P))$ (F) were evaluated on the test data part of the PDT corpus (the train and the development part were used for the lemmatizer training). The label Lem_H_G denotes the handcrafted lemmatizer with morphological guesser turned on (the guesser does not try to guess the correct lemma but only all possible morphological tags and, in addition, produces all presumably valid word forms for a given word). The labels Lem_PDT_oP and Lem_PDT_min denote the automatically trained lemmatizer using only OOV word patterns for lemmatization of all given words and the automatically trained lemmatizer using only prefix and suffix OOV patterns, respectively. No dictionary is used for lemma searching in the latter case and therefore this configuration can be seen as the minimal lemmatizer. In three last columns of the table are the results for the lemmatization of OOV words (words reported as unknown by Lem_H). There is only a small difference between both lemmatizers (Lem_H_G and Lem_PDT) recall (0.4 %) while the gap between precisions is much more significant (6.06 %). We will investigate the influence of these differences on the IR system in the next section.

3 IR Experiments

As mentioned before, our goal was to compare two approaches to the lemmatization on a real problem. Lemmatization was shown to improve the effectiveness of information retrieval in highly inflected languages (as is the Czech language) in earlier experiments [5,6].

3.1 Experimental Data

Our IR experiments were performed on the IR collection that was used in the Czech task of the Cross-Language Speech Retrieval track organized within the CLEF 2007 evaluation campaign [7]. This collection contains automatically transcribed

Table 3. Comparison of the lemmatizers

	Test data			OOV words		
	R[%]	P[%]	F	R[%]	P[%]	F
Lem_H	99.38	82.45	0.90	73.41	100.00	0.85
Lem_H_G	99.50	79.71	0.89	93.88	12.90	0.23
Lem_PDT	99.10	73.65	0.85	75.35	96.19	0.85
Lem_PDT_oP	81.77	98.33	0.89	73.26	99.09	0.84
Lem_PDT_min	75.79	98.59	0.86	72.67	99.69	0.84

spontaneous interviews (segmented by sliding a fixed-size window over the transcribed text into “documents”) and two sets of TREC-like topics – 29 training and 42 evaluation topics. Each topic consists of 3 fields – `<title>` (T), `<desc>` (D) and `<narr>` (N).

Both sets of topics were used for the experiments and two types of queries were created for each set of topics - first one from the terms from the T and D fields and the second one from all terms from the fields T, D and N. Stop words were omitted from all sets of query terms. The aforementioned mGAP measure that was used in the CLEF 2007 Czech task was used as an evaluation measure.

The correct lemma for our experiments is chosen based on the disambiguation of the output of the morphological analyzer by a tagger for the Lem_H_G lemmatizer whereas for the automatically trained lemmatizer the first supplied lemma is chosen.

3.2 IR System

Language modeling approach [8] was used as the information retrieval method for the lemmatizer evaluation, concretely the query likelihood method with an linear interpolation unigram language model of the document with an unigram language model of the whole collection. The idea of this method is to create a language model M_d from each document d and then for each query q to find the model which most likely generated the query, that means to rank the documents according to the probability $P(d|q)$. We use the Bayes rule: $P(d|q) = P(q|d)P(d)/P(q)$, where $P(q)$ is the same for all documents and the prior document probability $P(d)$ is uniform across all documents, so we can ignore both. We have left the probability of the query been generated by a document model $P(q|M_d)$, which can be estimated using the maximum likelihood estimate (MLE): $\hat{P}(q|M_d) = \prod_{t\in q} \frac{tf_{t,d}}{L_d}$, where $tf_{t,d}$ is the frequency of the term t in d and L_d is the total number of tokens in d. To deal with the sparse data for the generation of the M_d we use the mixture model between the document-specific multinomial distribution and the multinomial distribution of the whole collection M_c with interpolation parameter λ . So the final equation for ranking the documents according to the query is: $P(d|q) \propto \prod_{t\in q}(\lambda P(t|M_d) + (1-\lambda)P(t|M_c))$

3.3 Experiments Results

In the following text we compare the retrieval results of the two approaches to the lemmatization described above. For the case of automatically created lemmatizer we

Table 4. Comparison of mGAP score for lemmatized and non-lemmatized queries

test data	words	Lem_H_G	Lem_PDT	Lem_Ispell
train TD	0.0163	0.0270	0.0322	0.0280
train TDN	0.0164	0.0343	0.0364	0.0362
eval TD	0.0114	0.0220	0.0250	0.0200
eval TDN	0.0126	0.0274	0.0307	0.0243

have two sets of results - each for different lemmatizer training data. Table 4 shows the mGAP score for the two sets of test data (training, evaluation) and the two sets of terms (TD, TDN) as described in Sect. 3.1. Interpolation parameter λ was set to 0.5. The retrieval results for all three lemmatizers are significantly better than the result for non-lemmatized data (words) for all sets of queries and terms.

As can be seen from table 4, the retrieval results when compared with Lem_H_G lemmatizer are better for the Lem_PDT lemmatizer for both sets of queries and terms. For the Lem_Ispell (again compared with Lem_H_G) the results are better for the training set of queries and worse for the evaluation set. Because the retrieval performance of this IR system can differ for various levels of interpolation, we have run tests for few different settings of λ. The results are shown in tables 5 and 6, pretty similar course for all levels of interpolation can be seen there.

Table 5. Comparison of mGAP score for lemmatized queries of training set

term set	TD					TDN				
lemma / λ	0.1	0.25	0.5	0.75	0.9	0.1	0.25	0.5	0.75	0.9
Lem_H_G	0.0306	0.0290	0.0270	0.0261	0.0251	0.0392	0.0376	0.0343	0.0317	0.0295
Lem_PDT	0.0352	0.0343	0.0322	0.0298	0.0278	0.0396	0.0388	0.0364	0.0343	0.0307
Lem_Ispell	0.0328	0.0303	0.0280	0.0268	0.0255	0.0415	0.0397	0.0362	0.0329	0.0306
Lem_PDT_min	0.0326	0.0321	0.0305	0.0277	0.0264	0.0364	0.0345	0.0325	0.0305	0.0269
Lem_Ispell_min	0.0286	0.0274	0.0255	0.0231	0.0221	0.0394	0.0374	0.0347	0.0321	0.0296

Table 6. Comparison of mGAP score for lemmatized queries of evaluation set

term set	TD					TDN				
lemma / λ	0.1	0.25	0.5	0.75	0.9	0.1	0.25	0.5	0.75	0.9
Lem_H_G	0.0200	0.0212	0.0220	0.0222	0.0215	0.0255	0.0257	0.0274	0.0271	0.0260
Lem_PDT	0.0236	0.0243	0.0250	0.0252	0.0250	0.0281	0.0310	0.0307	0.0287	0.0271
Lem_Ispell	0.0193	0.0200	0.0200	0.0194	0.0198	0.0227	0.0234	0.0243	0.0243	0.0235
Lem_PDT_min	0.0186	0.0193	0.0197	0.0198	0.0195	0.0217	0.0215	0.0219	0.0215	0.0209
Lem_Ispell_min	0.0192	0.0199	0.0205	0.0204	0.0197	0.0178	0.0185	0.0178	0.0181	0.0168

3.4 Results evaluation

For the confirmation of our hypotheses, we ran several statistical significance tests. First, we claim that the retrieval results for the lemmatized data are better than the results for non-lemmatized data. The difference has shown to be statistically significant (with the significance level $\alpha = 0.01$) for all three tested lemmatizers when tested across all the query and terms sets and different settings of the retrieval method. The difference has also shown to be statistically significant when tested across the queries in one set for one setting of the IR method.

Then we tested automatically created lemmatizers against the manually created one. When tested across all the query and terms sets and different settings of IR method, the difference between Lem_H_G and Lem_PDT has shown to be statistically significant (with the significance level $\alpha = 0.01$) and the difference between Lem_H_G and Lem_Ispell has not shown to be statistically significant. When tested across queries in one set the difference for both automatically created lemmatizers has not shown to be statistically significant. We believe that is due to the large variance of the GAP score among the queries in the set and small number of queries. The Wilcoxon Matched-Pairs Signed-Ranks Test [9] was used for all tests. The last two rows in tables 5 and 6 show retrieval results for lemmatizers with minimal configuration (Lem_PDT_min, Lem_Ispell_min). The difference in the recall of the lemmatizers seems to affect the retrieval precision, but the result is still superior in comparison with using non-lemmatized data and is especially suitable for the memory efficient IR systems.

4 Conclusions and Future Work

The results achieved in experiments shown in Sec. 3.3 suggest that, when using the lemmatizer for the IR system purposes, there is no substantial difference in performance between manually and automatically created lemmatizer. Actually, the automatically created lemmatizer (Lem_PDT) even improved the retrieval performance within our experimental setting (as the gain in the mGAP score has been shown to be statistically significant for the IR paradigm and the test collection we have used – see Sect. 3.4). This result is especially promising in the prospect of development of IR systems for other languages since thanks to the existence of the Ispell resources for many languages, an acceptable lemmatizer can be easily built without any need of a manually created corpora or a handcrafted morphological analyzer (lemmatizer).

Just based on the presented experiments, it can not be said for sure what caused the observed performance gain. The first analysis of the results hints that the improvement could stem from the different approach to the lemmatization of some terms crucial for retrieving the relevant documents rather than from better overall precision and/or recall of the lemmatizer. More thorough examination of these causes and also a large-scale testing of these phenomenons using other information retrieval methods is a suitable matter for further work.

References

1. Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M.: Prague Dependency Treebank 2.0. Linguistic Data Consortium, Philadelphia, USA (2006)
2. Hajič, J., Hladká, B.: Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In: Proceedings of COLING-ACL Conference, Montreal, Canada, pp. 483–490 (1998)

3. Kanis, J., Müller, L.: Automatic Lemmatizer Construction with Focus on OOV Words Lemmatization. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 132–139. Springer, Heidelberg (2005)
4. Ispell dictionaries and rules files, http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html
5. Ircing, P., Müller, L.: Benefit of Proper Language Processing for Czech Speech Retrieval in the CL-SR Task at CLEF 2006. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 759–765. Springer, Heidelberg (2007)
6. Ircing, P., Psutka, J., Vavruška, J.: What Can and Cannot Be Found in Czech Spontaneous Speech Using Document-Oriented IR Methods UWB at CLEF 2007 CL-SR Track. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 712–718. Springer, Heidelberg (2008)
7. Ircing, P., Pecina, P., Oard, D.W., Wang, J., White, R.W., Hoidekr, J.: Information Retrieval Test Collection for Searching Spontaneous Czech Speech. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 439–446. Springer, Heidelberg (2007)
8. Ponte, J.M., Croft, W.B.: A language Modeling Approach to Information Retrieval. In: SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 275–281. ACM, New York (1998)
9. The Wilcoxon matched-pairs signed-ranks test, http://www.fon.hum.uva.nl/service/statistics/signed_rank_test.html

Evaluation of a Sentence Ranker for Text Summarization Based on Roget's Thesaurus

Alistair Kennedy[1] and Stan Szpakowicz[1,2]

[1] School of Information Technology and Engineering
University of Ottawa, Ottawa, Ontario, Canada
{akennedy,szpak}@site.uottawa.ca
[2] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. Evaluation is one of the hardest tasks in automatic text summarization. It is perhaps even harder to determine how much a particular component of a summarization system contributes to the success of the whole system. We examine how to evaluate the sentence ranking component using a corpus which has been partially labelled with Summary Content Units. To demonstrate this technique, we apply it to the evaluation of a new sentence-ranking system which uses *Roget's Thesaurus*. This corpus provides a quick and nearly automatic method of evaluating the quality of sentence ranking.

1 Motivation and Related Work

One of the hardest tasks in Natural Language Processing is text summarization: given a document or a collection of related documents, generate a (much) shorter text which presents only the main points. A summary can be generic – no restrictions other than the required compression – or query-driven, when the summary must answer a few questions or focus on the topic of the query. Language generation is a hard problem, so summarization usually relies on *extracting* relevant sentences and arranging them into a summary. While it is, on the face of it, easy to produce *some* summary, a *good* summary is a challenge, so evaluation is essential. We discuss the use of a corpus labelled with Summary Content Units for evaluating the sentence ranking component of a query-driven extractive text summarization system. We do it in two ways: directly evaluate sentence ranking using Macro-Average Precision; and evaluate summaries generated using that ranking, thus indirectly evaluating the ranking system itself.

The annual Text Analysis Conference (TAC; formerly Document Understanding Conference, or DUC), organized by The National Institute of Standards and Technology (NIST), includes tasks in text summarization. In 2005–2007, the challenge was to generate 250-word query-driven summaries of news article collections of 20–50 articles. In 2008–2009 (after a 2007 pilot), the focus has shifted to creating *update summaries* where the document set is split into a few subsets, from which 100-word summaries are generated.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 101–108, 2010.

```
<line>As opposed to the international media hype that surrounded last week's flight, with hundreds of journalists on site to capture the historic moment, Airbus chose to conduct Wednesday's test more discreetly.<annotation scu-count="2" sum-count="1" sums="0"><scu uid="11" label="Airbus A380 flew its maiden test flight" weight="4"/><scu uid="12" label="taking its maiden flight April 27" weight="3"/></annotation></line>
<line>After its glitzy debut, the new Airbus super-jumbo jet A380 now must prove soon it can fly, and eventually turn a profit.<annotation scu-count="0" sum-count="3" sums="14,44,57"/></line>
<line>"The takeoff went perfectly," Alain Garcia, an Airbus engineering executive, told the LCI television station in Paris.</line>
```

Fig. 1. Positive, negative and neutral sentence examples for the query "Airbus A380 – Describe developments in the production and launch of the Airbus A380"

1.1 Summary Evaluation

One kind of manual evaluation at DUC/TAC is a full evaluation of the *readability* and *responsiveness* of the summaries. Responsiveness tells us how good a summary is; the score should be a mix of grammaticality and content.

Another method of manual evaluation is *pyramid evaluation* [1]. It begins with creating several reference summaries and determining what information in them is most relevant. Each relevant element is called a Summary Content Unit (SCU), carried in text by a fragment, from a few words to a sentence. All SCUs are marked in the reference summaries and make up a so-called pyramid, with few frequent SCUs at the top and many rarer ones at the bottom. In the pyramid evaluation proper, annotators identify SCUs in peer summaries. The *SCU count* tells us how much relevant information a peer summary contains, and what redundancy there is if a SCU appears more than once. The *modified pyramid score* measures the recall of SCUs in a peer summary [2].

1.2 The SCU-Labelled Corpus

Pyramid evaluation supplies fully annotated peer summaries. Those are usually extractive summaries, so one can map sentences in them back to the original corpus [3]. Many sentences in the corpus can be labeled with the list of SCUs they contain, as well as the score for each of these SCUs and their identifiers. [3] reported that 83% of the sentences from the peer summaries in 2005 and 96% of the sentences from the peer summaries in 2006 could be mapped back to the original corpus. A dataset has been generated for the DUC/TAC main task data in years 2005–2009, and the update task in 2007.

We consider three kinds of sentences, illustrated in Figure 1. First, a *positive* example: its <annotation> tag shows its use in summary with ID 0, and lists two SCUs: with ID 11, weight 4, and with ID 12, weight 2. The second sentence – a *negative* example – has a SCU count of 0, but is annotated because of its use in summaries 14, 44 and 57. The third *unlabelled* sentence was not used in any summary: no annotation. The data set contains 19,248 labelled sentences from a total of 91,658 in 277 document sets.[1] The labelled data are 39.7% positive.

Parts of the SCU-labelled corpus have been used in other research. In [4], the 2005 data are the means for evaluating two sentence-ranking graph-matching algorithms

[1] There is one document set from each of the 2005–2007 main tasks, three sets from the 2007 update task and two sets each from the 2008–2009 main tasks.

for summarization. The rankers match the parsed sentences in the query with parsed sentences in the document set. For evaluation, summaries were constructed from the highest-ranked sentences. The sum of sentence SCU scores was the score for the summary. One problem with this method is that both labelled and unlabelled data were used in this evaluation, thus making the summary SCU scores a lower bound on the expected scores of the summary. Also the method does not directly evaluate a sentence ranker on its own, but rather in tandem with a simple summarization system.

In [5], an SVM is trained on positive and negative sentences from the 2006 DUC data and tested on the 2005 data. The features include sentence position, overlap with the query and others based on text cohesion. In [6], the SCU-labelled corpus is used to find a baseline algorithm for update summarization called Sub-Optimal Position Policy (SPP), an extension of Optimal Position Policy (OPP) [7]. In [8], the corpus from 2005–2007 is used to determine that summaries generated automatically tend to be query-biased (answer a query) rather than query-focused (select sentences to maximize overlap with a query).

2 Sentence Ranking

We compare a new method of sentence ranking against a variety of baselines, which we also describe here. The proposed method uses a function for measuring semantic distance between two terms, available through Open *Roget's*, a Java implementation of the 1911 *Roget's Thesaurus* ⟨rogets.site.uottawa.ca⟩. We also had access to a version with proprietary 1987 *Roget's* data, which allowed us to compare newer and older vocabulary. *Roget's* is a hierarchical thesaurus comprising 9 levels; words are always at the lowest level in the hierarchy.[2]

2.1 *Roget's* SemDist

The *SemDist* function based on *Roget's* was originally implemented for the 1987 version [9] but has recently also been made available for the 1911 version [10]. We use a modified version of *SemDist* pairs of words are scored 0..18, farthest to closest, where 18 is given when a word is compared with itself. The distance returned by *SemDist* is simply the edge distance between two words in *Roget's Thesaurus*, subtracted from 18. *SemDist* is used to generate a score *score*(*S*) indicating the similarity of a sentence *S* to the query *Q*. The distance between each word $w \in S$ is measured against each word $q \in Q$.

$$score(S) = \sum_{q \in Q} \max\left(SemDist(w, q) : w \in S\right)$$

score(*S*) ranks sentences by relevance to the query. This system can be implemented without *SemDist*: let a score be either 0 or 18. We also ran an experiment with this method – called Simple Match (SM) – and the methods using the 1911 and 1987 Thesauri. Stop words (we used the stop list from [11]) and punctuation are removed

[2] The 1911 version has around 100,000 words and phrases (60,000 of them unique), the 1987 version – some 250,000 (100,000 unique) words and phrases.

from both the queries and the sentences. This method tends to favour long sentences: a longer sentence has more chances of one of its words having a high similarity score to a given word in the query q_i. We see the effect of this tendency in an indirect evaluation by summary generation (Section 3.2).

2.2 Term Frequency – Inverse Sentence Frequency (*tf.isf*)

Term Frequency - Inverse Document Frequency (*tf.idf*) is widely used in document classification. We rank sentences, not documents, so we talk of Term Frequency – Inverse Sentence Frequency (*tf.isf*). The query is also treated as a single sentence (even if it has a few actual sentences). Again, stop words and punctuation are ignored. Cosine similarity is used to determine the distance between the query and each sentence. This is similar to what was done in [12].

2.3 Other Baselines

We include three baseline methods for comparison's sake. One is simply to rank sentences based on the number of words in them; referred to as *Length*. The second method is to order the sentences randomly; we label this method *Random*. The last method, *Ordered*, is to not bother ranking the sentences on any criteria: sentences are selected in the order in which they appear in the data set.

3 Evaluation and Results

We now discuss the evaluation of the systems from Section 2: *SemDist* 1987 and 1911, Simple Match and *tf.isf*, plus the three baseline methods. They will undergo two kinds of evaluation with the SCU-labelled corpus. In Section 3.2 we discuss an evaluation similar to what is performed in [4], but we exclude unlabelled sentences from the evaluation and only generate summaries with up to 100 words. We already noted a drawback: rather than directly evaluate the sentence ranker, this evaluates a ranker in tandem with a simple sentence-selection system.

We also need a method of determining how well a sentence ranker works on its own. To do this, in Section 3.1 we evaluate our ranked list of sentences using Macro-Average Precision. This will give us an overall score of how well the sentence ranker separates positive from negative sentences. We choose Macro-Average instead of Micro-Average, because the score each sentence receives depends on the query it is answering, so scores are not comparable between document sets. Again unlabelled sentences are excluded from this evaluation.

3.1 Direct Evaluation with Macro-average Precision

The calculation of average precision begins by sorting all the sentences in the order of their score. Next, we iterate through the list from highest to lowest, calculating the precision at each positive instance and averaging those precisions.

$$AveP = \sum_{r=1}^{N} Prec(r) \times Rec(r)$$

$Prec(r)$ is the precision up to sentence r, $Rec(r)$ – the change in recall [13]. The macro-average of the average precision is taken over all document sets, thus giving the macro-average precision, reported in Table 1.

Table 1. SCU Rankings for data

	SD-1911	**SD-1987**	**SM**	***tf.isf***	**Length**	**Random**	**Ordered**
Score	**0.572**	0.570	0.557	0.521	0.540	0.445	0.460

The differences between the systems are not very high: *SemDist* 1911 scores only 5.1% higher than *tf.isf*. The improvement of *SemDist* 1911 over the *Random* and *Ordered* baselines is more noticeable, but the *Length* baseline performs very well. Nonetheless, it can clearly be seen from these results that the two *Roget's SemDist*-based methods perform better than the others. There are a total of 277 document sets in the whole data set, which is a suitably high number for determining whether the differences between systems are statistically significant. A paired t-test shows that the two *SemDist* methods were superior to all others at $p < 0.01$, but the difference between the two *SemDist* methods was not statistically significant. The SCU-labelled corpus provides us with a way to prove that one sentence ranker outperforms another.

One possible problem with this evaluation approach is that it does not take sentence length into account. Long sentences are more likely to contain a SCU simply by virtue of having more words. An obvious option for evaluation would be to normalize these scores by sentence length, but this would actually be a different ranking criterion. Were we to modify the ranking criteria in this way, we would find that *tf.isf* outperforms *SemDist* 1911, 1987 and Simple Match. That said, favouring longer sentences is not necessarily a bad idea when it comes to extractive text summarization. Each new sentence in a summary will tend to jump to a different topic within the summary, to the detriment of the narrative flow. Selecting longer sentences alleviates this by reducing the number of places where the flow of the summary is broken.

All these sentence-ranking methods could be also implemented with *Maximal Marginal Relevance* [14], but once again that is simply a different ranking criterion and can still be evaluated with this technique.

3.2 Indirect Evaluation by Generating Summaries

A second evaluation focuses on indirectly evaluating sentence rankers through summary generation. We demonstrate this on a fairly simple summary evaluation system. To build the summaries, we take the top 30 sentences from the ranked list of sentences. Iterating from the highest-ranked sentence to the lowest in our set of 30, we try to add each sentence to the summary. If the sentence makes the summary go over 100 words, it is skipped and we move to the next sentence.

Table 2 shows the results. We report the total/unique SCU score, total/unique SCU count and the number of positive and negative sentences in the 277 summaries generated. Unique SCU score is probably the most important measure, because it indicates the total amount of unique information in the summaries.

Table 2. Total SCU scores and counts in all summaries

System	**Total Score**	**Unique Score**	**Total SCUs**	**Unique SCUs**	**Positive Sentences**	**Negative Sentences**
SemDist 1911	2,627	**2,202**	1,067	907	530	**396**
SemDist 1987	2,497	2,126	1,023	874	515	398
Simple Match	**2,651**	2,200	**1,083**	**923**	542	466
tf.isf	2,501	2,122	1,001	864	**572**	721
Length	1,336	1,266	678	567	286	258
Random	1,726	1,594	762	676	438	854
Ordered	1,993	1,709	855	741	491	771

Simple Match and *SemDist* 1911 perform better than the other methods, one leading in the total SCU score and one leading in the unique SCU score. That said, the difference in terms of SCU scores and SCU counts between systems is very small. The most significant differences can be seen in the number of positive and negative sentences selected. Since *tf.isf* does not favour longer sentences, it is natural that it would select a larger number of sentences. Counted as the percentage of *tf.isf*'s selection of positive sentences, the 1987 *SemDist* method, the 1911 method and Simple Match give 90%, 93% and 95% respectively. The difference in the number of negative sentences is more pronounced. The 1987 and 1911 *SemDist* methods have just 55% as many negative sentences as *tf.isf* while Simple Match methods has 65%. This sort of evaluation shows that the ratio of positive to negative sentences is the highest for the *SemDist*-based methods and Simple Match. This supports the findings in Section 3.1. In fact, the percentage of positive sentences selected by *SemDist* 1911, 1987 and Simple Match was 57%, 56% and 54% respectively, while *tf.isf* had only 44%.

This indirect evaluation showcases the downside of relying too heavily on sentence length. The *Length* baseline performs very poorly on almost every measure; even the *Random* baseline beats it on all but negative sentence count. The percentage of sentences selected using *Length* is about 42% of how many were selected using any of the methods which do not favour longer sentences (*tf.isf*, *Random* and *Ordered*), averaging just 1.96 sentences per summary. By comparison, Simple Match had about 78% and the two *SemDist* methods contained about 71% of the number of sentences used by *tf.isf*, *Random* and *Ordered*.

Our method of evaluating summary generation can also estimate redundancy in a summary by examining the number of total and unique SCUs. Both *SemDist* methods, Simple Match and *tf.isf* had about 85% as many unique SCUs as total SCUs. This is comparable to the baseline methods of *Length*, *Random* and *Ordered* which had 84%, 89% and 87% as many unique SCUs as total SCUs respectively. There is little redundancy in the summaries we generated, but redundancy is tied to summary length. As a quick experiment, we ran the 1911 *Roget's* SemDist function to generate summaries of sizes 100, 250, 500 and 1,000. We found that the percentage of unique SCUs dropped from 85% to 73% to 62% to 52%. This shows the need for such a redundancy-checking system, and clearly the SCU-based corpus can be a valuable tool for evaluating it.

4 Conclusions and Discussion

We have shown two methods of evaluating sentence ranking systems using a corpus partially labelled with Summary Content Units. This evaluation is quick and inexpensive, because it follows entirely from the pyramid evaluation performed by TAC. As long as TAC performs pyramid evaluation, the SCU-labelled corpus should grow without much additional effort. We have also shown that, despite their individual drawbacks, our direct and indirect methods of evaluating sentence selection complement each other. Evaluating sentence-ranking systems using Macro-Average Precision allows us to determine how good a sentence-ranking system is by taking every labelled sentence in the document set into account. Because of the large number of document sets available, it can be used to determine statistical significance in the differences between sentence-ranking systems. The drawback could be the favouring of simplistic methods of selecting longer sentences. The indirect evaluation through summary generation cannot be fooled by systems selecting long sentences. It also provides us with a means of measuring redundancy in summaries. Its drawbacks are that it only evaluates a sentence ranker as it is used for generating a summary in one particular way.

The fact that we ignore unlabelled sentences rather hurts this as an overall evaluation of a summarization system. Were a summarization system to select sentences in part because of their neighbours or location in a document, we could not guarantee that that sentence would be labelled. If, however, sentences are ranked and selected independent of its neighbours or location, then we can have a meaningful evaluation of the summarization system.

Our experiments showed that the *Roget's SemDist* ranker performed best when evaluated with Macro-Average Precision. Although the SCU scores from our evaluation in Section 3.2 did not show *Roget's SemDist* to have much advantage over *tf.isf* in terms of unique SCU weight, we found that it performed much better in terms of the ratio of positive to negative sentences. We also found that for sentence ranking the 1911 version of *Roget's* performed just as well as the 1987 version, which is unusual, because generally the 1987 version works better on problems using semantic relatedness [10].

Acknowledgments

NSERC and the University of Ottawa support this work. Thanks to Anna Kazantseva for many useful comments on the paper.

References

1. Nenkova, A., Passonneau, R.J.: Evaluating content selection in summarization: The pyramid method. In: HLT-NAACL, pp. 145–152 (2004)
2. Nenkova, A., Passonneau, R., McKeown, K.: The pyramid method: Incorporating Human Content Selection Variation in Summarization Evaluation. ACM Trans. Speech Lang. Process. 4 (2007)
3. Copeck, T., Inkpen, D., Kazantseva, A., Kennedy, A., Kipp, D., Nastase, V., Szpakowicz, S.: Leveraging DUC. In: HLT-NAACL 2006 - Document Understanding Workshop, DUC (2006)

4. Nastase, V., Szpakowicz, S.: A Study of Two Graph Algorithms in Topic-Driven Summarization. In: Proc. TextGraphs: 1st Workshop on Graph Based Methods for Natural Language Processing, pp. 29–32 (2006)
5. Fuentes, M., Alfonseca, E., Rodríguez, H.: Support Vector Machines for Query-Focused Summarization Trained and Evaluated on Pyramid Data. In: Proc. 45th Annual Meeting of the ACL, Poster and Demonstration Sessions, pp. 57–60 (2007)
6. Katragadda, R., Pingali, P., Varma, V.: Sentence Position Revisited: a Robust Light-Weight Update Summarization 'baseline' Algorithm. In: CLIAWS3 2009: Proc. Third International Workshop on Cross Lingual Information Access, pp. 46–52 (2009)
7. Lin, C.Y., Hovy, E.: Identifying topics by position. In: Proc. Fifth Conference on Applied Natural Language Processing, Morristown, NJ, USA, pp. 283–290. Association for Computational Linguistics (1997)
8. Katragadda, R., Varma, V.: Query-Focused Summaries or Query-Biased Summaries? In: Proc. ACL-IJCNLP 2009 Conference Short Papers, pp. 105–108 (2009)
9. Jarmasz, M., Szpakowicz, S.: Roget's Thesaurus and Semantic Similarity. In: Recent Advances in Natural Language Processing III. Selected papers from RANLP 2003. CILT, vol. 260, pp. 111–120. John Benjamins, Amsterdam (2004)
10. Kennedy, A., Szpakowicz, S.: Evaluating Roget's Thesauri. In: Proc. ACL 2008: HLT, Association for Computational Linguistics, pp. 416–424 (2008)
11. Jarmasz, M., Szpakowicz, S.: Not As Easy As It Seems: Automating the Construction of Lexical Chains Using Roget's Thesaurus. In: Proc. 16th Canadian Conference on Artificial Intelligence (AI 2003), Halifax, Canada, pp. 544–549 (2003)
12. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-Based Summarization of Multiple Documents. Inf. Process. Manage. 40, 919–938 (2004)
13. Zhu, M.: Recall, Precision and Average Precision. Technical Report 09, Department of Statistics & Actuarial Science, University of Waterloo (2004)
14. Carbonell, J., Goldstein, J.: The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In: Research and Development in Information Retrieval, pp. 335–336 (1998)

Real Anaphora Resolution Is Hard

The Case of German

Manfred Klenner, Angela Fahrni, and Rico Sennrich

Institute of Computational Linguistics, Binzmuehlestrasse 14, CH-8050 Zurich
{klenner,sennrich}@cl.uzh.ch,
angela.fahrni@swissonline.ch

Abstract. We introduce a system for anaphora resolution for German that uses various resources in order to develop a real system as opposed to systems based on idealized assumptions, e.g. the use of true mentions only or perfect parse trees and perfect morphology. The components that we use to replace such idealizations comprise a full-fledged morphology, a Wikipedia-based named entity recognition, a rule-based dependency parser and a German wordnet. We show that under these conditions coreference resolution is (at least for German) still far from being perfect.

1 Introduction

Anaphora and coreference resolution is a central task in the course of text understanding. At the sentence level, the resolution of anaphora is a prerequisite for semantic interpretation and at the text level it contributes to coherence and discourse structure. Although a lot of work has been done in the field of coreference resolution, real systems carrying out full fledged coreference resolution including pronominal and nominal anaphora are the exception. Most of the time, researchers (including the authors of this paper) try to cut away the complexity of the task and work under idealized conditions. One can find this kind of simplifications in almost every paper presented at renowned international conferences. Among the idealization, the following are the most prominent:

1. perfect anaphoricity determination (i.e. true mentions only)
2. perfect parse trees
3. perfect functional information
4. perfect morphological analysis
5. perfect named-entity recognition

The most unrealistic and most simplifying idealization is to use true mentions (1) instead of all noun phrases (henceforth 'markables'). True mentions are those markables that are—according to the gold standard—part of a coreference chain. The majority of noun phrases in a text, however, are not in a coreference set. The determination whether a NP is anaphoric (i.e. a true mention) or not is a demanding problem, the so called anaphoricity classification problem. There are a few systems that incorporate anaphoricity classification, the majority of systems leaves this as an implicit task to the

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 109–116, 2010.

anaphora resolution component. Separate anaphoricity classification has not (really) proven to be more successful than its implicit counterpart. Anaphoricity determination of markables is a non-trival task and cutting it away makes a system an artificial one.

Syntactic information in form of parse trees is used in state of the art systems in a number of ways. Since most of the approaches (including ours) cast anaphora resolution as a (pairwise) classification task, features are needed. Among them are e.g. the depth of embedding of a markable, the part of speech of the head of a markable and even information related to intrasentential binding constraints (*c*-command). Working with idealized syntactic information pushes performance at unrealistic heights.

One of the most discriminative information is functional, namely grammatical roles. For example, parallelism of grammatical functions of a pronoun and its antecedent candidate is a powerful feature. Fortunately, dependency parsers are quite good in the recognition of grammatical functions (a subclass of dependency labels). Thus, this kind of idealization is less serious.

Especially in medium and highly inflectional languages such as German, morphological information establishes a powerful filter. E.g. personal pronouns must unify in person, number and gender. One can get rid of all pairs that do not fulfill this condition. This reduces the number of training examples, and thereby improves the quality of the classifier (by removal of safe negative examples).

Finally, named entity recognition is crucial for coreference resolution since—at least in newspaper texts—persons, groups and institutions play an important role. They are very likely to be referred to by pronouns or nominal anaphora. To know that a markable is e.g. a person helps the classifier a lot. Again, perfect information obscures the quality of a system for real applications.

There are other dimensions that prevent current systems from really being useful. To mention but one: there are performance problems arising from theoretically interesting but rather time consuming approaches, e.g. coreference resolution on the basis of integer linear programming (ILP), cf. e.g. [1]. It is appealing to have the means to express global constraints (e.g. transitivity of the anaphoric relation as a means to propagate binding constraints within a coreference set). But transitivity expressed as ILP constraints yields (at least for medium and longer texts) ten thousands of equations, which is not only a crude but also a time-consuming method to capture a simple property of the anaphoric relation.

We are not saying that experiments under idealized conditions are totally in vain. We are just arguing that it doesn't help a lot to tune a system on the basis of gold standard information if one intends to switch to a real-world system. One never foresees the amount of noise that is introduced by real components.

In this article, we introduce a realistic system for coreference resolution for German and describe its various components. We discuss our filter-based approach to pairwise classification, give empirical results and discuss the reason for the drop of performance from an idealized setting to a real world setting. We start by describing our filter-based approach to pairwise classification and the features we are using for machine learning. They are derived on the basis of real-word preprocessing components.

2 Our Filter-Based Approach

It is common practice to cast anaphora resolution as pairwise classification. Systems differ in the features they use, but also in their training procedures (fixed window of n sentences, Soon-style flexible window) and the kind of coreference clustering (best-first, closest-first, aggressive merging) they do in order to merge positively classified pairs into a partition of coreference sets.

In a former paper we have argued that coreference clustering based on the so-called Balas order coupled with intensional constraints to ensure consistency of coreference sets performs best [1]. In this paper, we concentrate on the features, their derivation and their quality. We do not discuss problems of coreference clustering. Just one hint why coreference clustering improves coreference resolution. The local perspective on pairs bears the danger of implicitly incompatible markables. Take the following markable chain: 'Hillary Clinton ... she ... Angela Merkel'. 'she' is compatible with 'Hillary Clinton', 'Angela Merkel' is compatible with 'she', but 'Merkel' and 'Clinton' are incompatible. Since transitivity is outside the scope of a pairwise classifier, it might classify both compatible pairs as positive without noticing that this leads to an implicit inconsistency.

Our system is filter-based, that is, only those pairs are considered as candidates that pass all filters. We have morphological, syntactic and semantic filters.

The morphological filters refer to person, number and gender. Personal pronouns must unify in each of them, while possessive pronouns only unify in person and gender, e.g. 'Er hat seine Brüder getroffen' ('He^i has met his^i brothers'), but not in number. 'seine' ('his') is plural, 'Er' ('He') is singular. Nominal anaphora in German only unify in number (and trivially in person), but not necessarily in gender ('Der Weg$^i_{masc}$ ist lang. Ich bin diese Strecke$^i_{fem}$ ...'). Each of these cases is covered by a rule and there are some rules for special cases, e.g. the rule for reported speech, where a third person pronoun is coreferent with a first person pronoun, e.g. 'Er sagte, ich ...' ('He said: I ...').

Among the syntactic filters, the subclause filter is the most prominent. It can be used to operationalize binding constraints and helps to reduce the amount of negative pairs. The constraint here is: two personal pronouns (or nouns) in the same subclause cannot be coreferent ('Sie^i gibt ihr^j das Buch', where $i \quad j$; 'She^i gives her^j the book'—in English, a reflexive pronoun is necessary to establish coreference). With possessive pronouns this is different, a possessive pronoun and its antecedent might be in the same subclause. For reflexive pronouns the antecedent even should be in the same subclause, but there are exceptions (sentences where the reflexive pronoun is not anaphoric at all).

Semantic filters are based on GermaNet [2], the German wordnet. Two nominal markables must be semantically compatible, which means that they must be both e.g. animate or inanimate, or stand in a hyponym or synonym relation. If one of the markables is not in GermaNet, the pair does not pass the filter (reducing recall). We have also experimented with selectional restrictions available from verb frames. If a personal pronoun fills e.g. the subject slot of a verb, semantic information becomes available by the selectional restriction of the verb slot (e.g. the subject of 'to sleep' is animate, neglecting metaphorical usages). This way, the number of valid candidate antecedents (noun phrases that are of type animate) can be further restricted.

We strive to integrate as much linguistic knowledge as possible into the filters. Alternatively, one could use this kind of linguistic knowledge as a feature. But our experiments have shown that a filter based approach is more reliable. There are only a few exceptions of these regularities (at least at the morphological and syntactic level). It's better to erroneously filter such pairs out as to let everything pass.

Any pair that has passed all filters gets classified by a machine learning programme. We use the memory-based learner TiMBL [3] as a classifier. This is done on the basis of following features:

- distance in sentences
- distance in markables
- part of speech of the heads (tagger)
- grammatical functions (parser)
- parallelism of grammatical functions (parser)
- salience of the grammatical functions of the heads (see below)
- depth of embedding of the heads (parser)
- whether an NP is definite or not (Gertwol)
- the semantic class (GermaNet)
- whether an NP is animate or not (GermaNet)
- whether the markables are in the same subclause (parser)

Salience of a grammatical function is estimated (on the basis of the training set) in the following way: the number of cases a grammatical function realizes a true mention divided by the total number of true mentions (it's the conditional probability of a grammatical function given an anaphor). The function 'subject' is the most salient function followed by 'direct object'.

3 System Components

The preprocessing step prior to pair-wise classification of anaphora candidates is crucial, since it produces the features used to describe the markables and thus indirectly determines the quality of the classifier. Fortunately, we have high performance tools available: the TreeTagger, GermaNet, Gertwol and Pro3GresDe (the parser). After tokenization and tagging the morphological analysis takes place. We use Gertwol, a commercial system based on two-level morphology. Gertwol is fast and also carries out noun decomposition which is rather useful, since in German compounds are realized as single wordforms, e.g. Computerexperte ('computer expert'). Compounds (which are quite frequent in German) might become very complex, but often the head of the compound is sufficient to semantically classify the whole compound via GermaNet. For instance, 'Netzwerkcomputerexperte' ('expert for network computers') is an expert and, thus, is animate. Gertwol decomposes the compound and the head can be classified with the aid of GermaNet. The other important task of Gertwol is to determine the number, person and gender information of a word. Unfortunately, ambiguity rate is high, since e.g. some personal pronouns are highly ambiguous. For instance, the German pronoun 'sie' ('she') might be singular/feminine or plural (without gender restriction). The pronoun 'ich' does not impose any gender restrictions and moreover often refers (in reported speech) to a speaker which is third person.

3.1 Named-Entity Recognition

Our Named-Entity Recognition (NER) is pattern-based, but also makes use of extensive resources. We have a large list of (international) first names (53,000) where the gender of each name is given. From Wikipedia we have extracted all multiword article names (e.g. ’Berliner Sparkasse’, a credit institute from Berlin) and, if available, their categories (e.g. ’Treptower Park’ has ’Parkanlage in Berlin Bezirk Treptow-Köpenick’ as its category tree; ’Parkanlage’ being the crucial information’).

The pattern-based NER uses GermaNet and Wikipedia and the information of the POS tagger. For instance, ’Grünen Bewegung Litauens’ is a multiword named entity. ’Litauens’ is genitive, thus it is not the head of the noun phrase, ’Bewegung’ (here: ’group’) is the head, so the whole compound denotes a group of people not a country. Since ’Grünen’ is an adjective in initial caps (which is unusual), it is considered as part of the name.

Our parser takes advantage of NER, since it reduces ambiguity and grouping problems.

3.2 Pro3gresDe: The Parser

Pro3GresDe is a hybrid dependency parser for German that is based on the English Pro3Gres parser [4]. It combines a hand-written grammar and a statistical disambiguation module trained on part of the TüBa-D/Z treebank [5].[1] This hybrid approach has proven especially useful for the functional disambiguation of German noun phrases. While the function of noun phrases is marked morphologically in German, many noun phrases are morphologically ambiguous, especially named entities. We use both morphological unification rules and statistical information from TüBa-D/Z (i.e. data about possible subcategorisation frames of verbs) to resolve functional ambiguities. We have shown that this approach performs better at functionally disambiguating noun phrases than purely statistical parsers.

The parser give access to the following features: e.g. grammatical function, depth of embedding, subclause information.

4 Empirical Evaluation

We have evaluated our base system only, i.e. without our clustering method described in [1]. It’s the baseline performance drop that we are interested in. The performance drop is measured in terms of save (gold standard) versus noisy (real-world components) morphological, functional and syntactic information. The gold standard information stems from the TüBa-D/Z treebank (phrase structure trees, topological fields, head information, morphology) which also is annotated with coreference links [7]. Our experiments are restricted to nominal anaphora and personal pronouns, i.e. we exclude the very simple cases of reflexive and relative pronouns, but also possessive pronouns, since we are focusing on the most demanding classes.

[1] For a full discussion of Pro3GresDe, see [6].

We have run the system with all markables and without any gold standard information (see Tab. 1). The f-measure of these runs (5-fold cross validation) is 58.01%, with a precision of 70.89% and a recall of 49.01%. The performance is low because recall is low. Precision on the other hand is good. The recall is low, since our filters for nominal anaphora are quite restrictive (fuzzy string match, GermaNet hyponomy and synonymy restrictions). Most of the false negatives stem from such filtered out nominal pairs. Refining our filters for nominal anaphora would clearly help to improve recall. Nominal anaphora are, however, the most challenging part of coreference resolution. Another reason for low recall is: we are working with a fixed window of 3 sentences in order to limit the number of candidate pairs. Only named-entities are allowed to refer back further than 3 sentences, but not personal pronouns and normal nouns. This way, we miss some long distance anaphoric relations. Our experiments have, however, shown that it is better to restrict the search than to generate any reachable pairs: performance drops to a great extent the larger the window.

Table 1. Performance Drop

	gold standard info	- morphological	- functional	- subclause (=real)
F-measure	61.49%	59.01%	58.20%	58.01%
Precicion	68.55%	69.78%	69.12%	70.89%
Recall	55.73%	51.12%	50.56%	49.01%

If we take gold standard information, especially perfect morphology, perfect syntax and perfect functional information, the f-measure value is 61.49%, about 3.5% above the real-world setting. Precision drops: 68.55%, but recall significantly increases to 55.73%. Thus, the reason for performance increase is the increase of recall. How can we explain it? Let us first see how the different gold standard resources contribute to this increase. If we turn grammatical functions from 'parser given' to 'gold standard given', the increase on the baseline is small: f-measure raises from 58.01% to 58.20%. Our dependency parser is good enough to almost perfectly replace gold standard information. The same is true with syntactic information concerning the depth of embedding and subclause detection. Here as well, only a small increase occurs: the f-measure is 59.01%. But if we add perfect morphology, an increase of 3.5% pushes the results to the final 61.49%.

The reason for the increase in recall (and f-measure) is our filter-based method. Only those pairs are generated that pass the filter. If the morphology is noisy, pairs erroneously might pass the filter and others pairs erroneously do not pass the filter. The first one spoils precision the second hampers recall.

We were quite surprised that the replacement of syntactic and functional information by real components was not the problem. Morphology is (mainly) responsible for the drop. See the next section for a comparison of our results with previous work.

5 Related Work

The work of [8] is a prototypical and often reimplemented machine-learning approach in the paradigm of pair-wise classification. Our system has a similar architecture, and our features do overlap to a great extent.

Work on coreference resolution for German is rare, most of it uses the coreference annotated treebank TüBa-D/Z. [9] uses a maximum entropy model for nominal anaphora resolution, his major insight is that if information from GermaNet is available then it outperforms the statistical model. We took this finding seriously and have tried to use Wikipedia to complement GermaNet (we map Wikipedia multiword items via Wikipedia categories to GermaNet classes). [10] introduce anaphora resolution (only pronouns) on the basis of a former version of the TüBa-D/Z. They also work with TiMBL. Their results are based on gold standard information and are – compared to subsequent work, cf. [11], where also gold standard information was utilized – surprisingly high (f-measure 73.40% compared to 58.40%).

A study concerning the influence of different knowledge sources and preprocessing components on pronoun resolution was carried out by [12]. [13] and [1] are concerned with the consistency of coreference sets using idealized input from the TüBa-D/Z treebank.

6 Conclusion

We have introduced a system for coreference resolution that makes extensive use of non-statistical resources (rule-based dependency parsing, a German wordnet, Wikipedia, two-level morphology) but at the same time is based on a state of the art machine learning approach. The system is not subject to any idealized assumptions related to the various preprocessing steps (i.e. no gold standard information is used), its empirical performance is, thus, not breath-taking. This is, however, not an embarrassing flaw. Rather, we think it is time to move away from idealized prototypes to assessing the performance of coreference resolution under real-world conditions.

We have shown that the performance drop, at least for coreference resolution in German, is mainly based on the morphological ambiguity introduced by replacing perfect morphological descriptions with the output of a real morphological analyzer. Most surprising to us was the finding that using a parser instead of gold standard information only had a small negative effect on the results.

Acknowledgments. Our project is funded by the Swiss National Science Foundation (grant 105211-118108).

References

1. Klenner, M., Ailloud, E.: Optimization in Coreference Resolution Is Not Needed: A Nearly-Optimal Zero-One ILP Algorithm with Intensional Constraints. In: Proceedings of the EACL (2009)

2. Hamp, B., Feldweg, H.: GermaNet—a Lexical-Semantic Net for German. In: Proc. of ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications (1997)
3. Daelemans, W., Zavrel, J., van der Sloot, K., van den Bosch, A.: TiMBL: Tilburg Memory-Based Learner (2004)
4. Schneider, G.: Hybrid Long-Distance Functional Dependency Parsing. Doctoral Thesis, Institute of Computational Linguistics, Univ. of Zurich (2008)
5. Telljohann, H., Hinrichs, E.W., Kübler, S.: The TüBa-D/Z Treebank: Annotating German with a Context-Free Backbone. In: Proc. of the Fourth Intern. Conf. on Language Resources and Evaluation, Lisbon, Portugal (2004)
6. Sennrich, R., Schneider, G., Volk, M., Warin, M.: A New Hybrid Dependency Parser for German. In: Proc. of the German Society for Computational Linguistics and Language Technology 2009 (GSCL 2009), Potsdam, Germany, pp. 115–124 (2009)
7. Naumann, K.: Manual for the Annotation of Indocument Referential Relations. Electronic document, (2006), http://www.sfs.uni-tuebingen.de/tuebadz.shtml
8. Soon, W., Ng, H., Lim, D.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics 27(4), 521–544 (2001)
9. Versley, Y.: A Constraint-Based Approach to Noun Phrase Coreference Resolution in German Newspaper Text. In: Konferenz zur Verarbeitung Natürlicher Sprache, KONVENS (2006)
10. Hinrichs, E., Filippova, K., Wunsch, H.: A Data-driven Approach to Pronominal Anaphora Resolution in German. In: Proc. of RANLP (2005)
11. Wunsch, H., Kübler, S., Cantrell, R.: Instance Sampling Methods for Pronoun Resolution. In: Proc. of RANLP, Borovets, Bulgaria (2009)
12. Schiehlen, M.: Optimizing Algorithms for Pronoun Resolution. In: Proceed. of the 20th International Conference on Computational Linguistics (2004)
13. Klenner, M., Ailloud, E.: Enhancing Coreference Clustering. In: Johansson, C. (ed.) Proc. of the Second Workshop on Anaphora Resolution (WAR II), Bergen, Norway. NEALT Proceedings Series, vol. 2 (2008)

Event-Time Relation Identification Using Machine Learning and Rules

Anup Kumar Kolya[1], Asif Ekbal[2], and Sivaji Bandyopadhyay[1]

[1] Department of Computer Science and Engineering, Jadavpur University,
Kolkata-700032, India
anup.kolya@gmail.com, sivaji_cse_ju@yahoo.com

[2] Department of Computational Linguistics, Heidelberg University,
Heidelberg-69120, Germany
ekbal@cl.uni-heidelberg.de, asif.ekbal@gmail.com

Abstract. Temporal information extraction is a popular and interesting research field in the area of Natural Language Processing (NLP). In this paper, we report our works on temporal relation identification within the TimeML framework. We worked on TempEval-2007 Task B that involves identification of relations between events and document creation time. Two different systems, one based on machine learning and the other based on handcrafted rules, are developed. The machine learning system is based on Conditional Random Field (CRF) that makes use of only some of the features available in TimeBank corpus in order to infer temporal relations. The second system is developed using a set of manually constructed handcrafted rules. Evaluation results show that the rule-based system performs better compared to the machine learning based system with the precision, recall and F-score values 75.9%, 75.9% and 75.9%, respectively under the strict evaluation scheme and 77.1%, 77.1% and 77.1%, respectively under the relaxed evaluation scheme. In contrast, CRF based system yields precision, recall and F-score values 74.1%, 73.6% and 73.8%, respectively under the strict evaluation scheme and 75.1%, 74.6% and 74.8%, respectively under the relaxed evaluation scheme.

Keywords: temporal relation identification, rule-based approach, conditional random field, TempEval-2007 Task B.

1 Introduction

Temporal information extraction is, nowadays, a popular and interesting research area of Natural Language Processing (NLP). Generally, events are described in different newspaper texts, stories and other important documents where events happen in time and the temporal location and ordering of these events are specified. Text analysis, among other capabilities, clearly requires identifying events described in a text and locating these in time. This is also important in a wide range of NLP applications that include temporal question answering, machine translation and document summarization. In TempEval-2007 [1], the three types of event-time temporal relations were considered: Task A (relation between the events and times within the same sentence), Task B (relation between events and document creation times) and Task C (relation between

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 117–124, 2010.

verb events in adjacent sentences). In each of these tasks, system attempts to annotate appropriate pairs with one of the following relations: BEFORE, BEFORE-OR-OVERLAP, OVERLAP, OVERLAP-OR-AFTER, AFTER or VAGUE. The participating teams were instructed to find all temporal relations of these types in a corpus of news-wire documents.

In the literature, temporal relation identification based on machine learning approaches can be found in Boguraev et el. [2], Mani et al. [3], Chambers et al. [4] and some of the TempEval-2007 participants [1]. Most of these works tried to improve classification accuracies through feature engineering. The performance of any machine learning based system is often limited by the amount of available training data. Mani et al. [3] introduced a temporal reasoning component that greatly expands the available training data. The training set was increased by a factor of 10 by computing the closure of the various temporal relations that exist in the training data. They reported significant improvement of the classification accuracies on event-event and event-time relations. However, this has two shortcomings, namely feature vector duplication caused by the data normalization process and the unrealistic evaluation scheme. The solutions to these issues are briefly described in [5]. In TempEval-2007 task [1], a common standard dataset was introduced that involves three temporal relations. The participants reported F-scores for event-event relations ranging from 42% to 55% and for event-time relations from 73% to 80%. Unlike [3] and [5], event-event temporal relations are not discourse-wide (i.e., any pair of events can be temporally linked) in TempEval-2007. Here, the event-event relations are restricted to events within two consecutive sentences. Thus, these two frameworks produce highly dissimilar results for solving the problem of temporal relation classification.

In order to apply various machine learning algorithms, most of the authors formulated temporal relation as an event paired with a time or another event and translated these into a set of feature values. Some of the popularly used machine learning techniques were Naive-Bayes, Decision Tree (C5.0), Maximum Entropy (ME) and Support Vector Machine (SVM). Machine learning techniques alone cannot always yield good accuracies. To achieve reasonable accuracy, some researchers [6] used hybrid approach, where rule-based component was combined with machine learning.

In our present work, we have developed two different models, one based on machine learning and the other based on handcrafted rules, to solve the problem of Task B. The main problem of Task B at TempEval-2007 [1] was to find out the relations between the events and document creation time. The machine learning based system is developed using the well known statistical algorithm, CRF. Here, the task of temporal relation identification is considered as a pair-wise classification problem, where each event/time pair is assigned one of the TempEval relation classes (BEFORE, AFTER, etc.). Event/time pairs are encoded using syntactically and semantically motivated features present in the TimeBank corpus. These features have been automatically extracted from the training corpus and used to train a CRF framework. It is to be noted that we have only used some of the features available in the training corpus. The rule-based system is based on a set of manually constructed handcrafted rules. Evaluation results show that the rule-based system performs better compared to the machine-learning based system with the improvement of 2.1% F-score under the strict evaluation scheme and 2.3% F-score under the relaxed evaluation scheme.

2 Conditional Random Field Based Approach

At first, we use machine learning for solving the problem of Task B. We consider the temporal relation identification task as a pair-wise classification problem in which the target pairs-a TIMEX3 tag and an EVENT-are modeled using CRF, which can include arbitrary set of features and still can avoid overfitting in a principled manner. Task B at TempEval-2007 holds temporal relations identification between event expressions and Document Creation Time (DCT). The events and temporal expressions (TEs) were annotated in the source in accordance with the TimeML standard [7]. The set of temporal relations to be predicted includes: OVERLAP, BEFORE, AFTER, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE. Furthermore, only event expressions that occur within the ETL are considered.

2.1 Conditional Random Field

Conditional Random Fields (CRFs) [8] are undirected graphical models, a special case of which corresponds to conditionally trained probabilistic finite state automata. Being conditionally trained, these CRFs can easily incorporate a large number of arbitrary, non-independent features while still having efficient procedures for non-greedy finite-state inference and training.

CRF is used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence $s =< s_1, s_2, \ldots, s_T >$ given an observation sequence $o =< o_1, o_2, \ldots, o_T >$ is calculated as:

$$P_\wedge(s|o) = \frac{1}{Z_o} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)\right),$$

where, $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight λ_k, is to be learned via training. The values of the feature functions may range between $-\infty, \ldots + \infty$, but typically they are binary. To make all conditional probabilities sum up to 1, we must calculate the normalization factor,

$$Z_o = \sum_{s} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)\right),$$

which as in HMMs, can be obtained efficiently by dynamic programming.

To train a CRF, the objective function to be maximized is the penalized log-likelihood of the state sequences given the observation sequences:

$$L_\wedge = \sum_{i=1}^{N} \log\left(P_\wedge(s^{(i)}|o^{(i)})\right) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2},$$

where $\{< o^{(i)}, s^{(i)} >\}$ is the labeled training data. The second sum corresponds to a zero-mean, σ^2 -variance Gaussian prior over parameters, which facilitates optimization by making the likelihood surface strictly convex. Here, we set parameters λ to maximize the penalized log-likelihood using Limited-memory BFGS [9], a quasi-Newton method that is significantly more efficient, and which results in only minor changes in accuracy due to changes in λ.

In general, CRFs can take any value between $-\infty, \ldots + \infty$, although binary values are traditional. A feature function $f_k(s_{t-1}, s_t, o, t)$ has a value of 0 for most cases and is only set to be 1, when s_{t-1}, s_t are certain states and the observation has certain properties. We have used the C^{++} based CRF^{++} package [1], a simple, customizable, and open source implementation of CRF for segmenting or labeling sequential data.

2.2 Features

We use the gold-standard TimeBank features for events and times for training the CRF. In the present work, we mainly use the various combinations of the following features:

1. Part of Speech (POS) of event terms: It denotes the POS information of the event. The features values may be either of ADJECTIVE, NOUN, VERB, and PREP.
2. Event Tense: This feature is useful to capture the standard distinctions among the grammatical categories of verbal phrases. The tense attribute can have values, PRESENT, PAST, FUTURE, INFINITIVE, PRESPART, PASTPART, or NONE.
3. Event Aspect: It denotes the aspect of the events. The aspect attribute may take values, PROGRESSIVE, PERFECTIVE and PERFECTIVE PROGRESSIVE or NONE.
4. Temporal Relation between the Document Creation Time and the Temporal Expression in the target sentence: The value of this feature could be “greater than”, “less than”, “equal”, or “none”.

3 Rule Based Approach

We manually design a set of rules by analyzing the various features available in the TimeBank corpus in order to infer the relation betweens events and DCTs. There are some exceptions to these rules. However, a rule is used if it is found to be correct most of the time throughout the training data. Below, we list our manually identified set of six rules:

1. If ’past’ is the feature value of ’Tense’ then the relation type between the event and DCT is set to ’before’.
2. If ’aspect’ is ’perfect’ and ’tense’ is ’present’ then the relation is set to ’before’.
3. If ’tense’ is ’none’ and ’POS’ is ’noun’ then the relation is set to ’before’.
4. If ’class’ is ’reporting’ then the relation is set to ’before’.
5. If ’tense’ is ’present’ and ’pos’ is ’verb’ then the relation is set to ’overlap’.
6. Relation is set to ’after’ for all other conditions.

[1] http://crfpp.sourceforge.net

The first four steps (designated as Rule Group 1) contribute for the identification of BEFORE relation, fifth one (designated as Rule Group 2) helps to identify 'OVERLAP' relation and the sixth rule (designated as Rule Group 3) is helpful to identify the 'AFTER' relation between the event and DCT. Support and confidence of each rule as computed from the training data is shown below:

1. **Rule Group 1:** Total number of links in training data= 331, Number of links where BEFORE relation holds= 210, Number of links correctly identified by our Rule 1=170, Support of Rule 1= 170/331=51.36%, Confidence of Rule 1= 170/210=81.00%
2. **Rule Group 2:** Total number of links in training data= 331, Number of links where OVERLAP relation holds= 61, Number of links correctly identified by our Rule 2=32, Support of Rule 2= 32/331=9.67%, Confidence of Rule 2= 32/61=52.45%
3. **Rule Group 3:** Total number of links in training data= 331, Number of links where AFTER relation holds= 60, Number of links correctly identified by our Rule 3=31, Support of Rule 3= 31/331=9.66%, Confidence of Rule 3= 31/60=52.44%

It is to be noted that support and confidence of Rule Group 1 is very high compared to other two groups as 75% of the relations in the training data are BEFORE relation type.

4 Experimental Result and Discussions

We develop a number of models of CRF based on the features included into it. A feature vector consisting of the available features as described in Section 2.2 is extracted for each <event, DCT> pair in the TimeBank corpus. Now, we have a training data in the form (w_i,t_i), where, w_i is the i^th pair along with its feature vector and t_i is its corresponding TempEval relation class. Models are built based on the training data and the feature template. The procedure of training is summarized below:

1. Define the training corpus, C.
2. Extract the <event, DCT> relations from the training corpus.
3. Create a file of candidate features, including lexical features derived from the training corpus.
4. Define a feature template.
5. Compute the CRF weights λ_k for every f_K using the CRF toolkit with the training file and feature template as input.
6. Derive the best feature template depending upon the performance.
7. Select the best feature template obtained from Step 6.
8. Retrain the CRF model

During evaluation, we obtain the highest performance for the following feature template as shown in Table 1. The test data consists of 20 articles from TimeBank [10]. The performance is assessed with three evaluation metrics (precision, recall, F-score) and two scoring schemes (strict, relaxed), as was used in the TempEval-2007 shared task [1]. The strict scoring scheme counts only exact matches, while the relaxed one gives credit to partial semantic matches too.

Table 1. Best Feature Template of the CRF based System

w_{i-2}
w_{i-1}
w_i
w_{i+1}
w_{i+2}
Combination of w_{i-1} and w_i
Combination of w_i and w_{i+1}
Dynamic output tag (t_i) of the previous pair
Feature vector of w_i of other features

Evaluation results with different feature representations are reported in Table 2 for CRF. Results show that the system performs better with the context of size five (i.e., previous two, current and the next two <event, DCT> pairs), tense and aspect features. It shows the precision, recall and F-score values of 71.4%, 71.0% and 71.2%, respectively under the strict evaluation scheme and 71.8%, 71.3% and 71.5%, respectively under the relaxed evaluation scheme.

Table 2. Evaluation result of CRF using different feature combinations

Feature Combination	Strict			Relaxed		
	precision	recall	F-score	precision	recall	F-score
Context, Tense, POS	70.8 %	70.3 %	70.5 %	71.1 %	70.6 %	70.9 %
Context, Aspect, POS	57.5 %	57.1 %	57.3 %	57.8 %	57.4 %	57.6 %
Context, Tense, Aspect	71.4 %	71.0 %	71.2 %	71.8 %	71.3 %	71.5 %

The overall evaluation results of the system are presented in Table 3. It shows the results of the *baseline* model, CRF based system as well as the rule-based system. The *baseline* model is developed based on the most frequent temporal relation encountered in the training data for the task. In the case of task B, the most frequent temporal relation present in the training data is BEFORE. Results show that the CRF based system performs better than the *baseline* model with the margins of 16.7% F-score in the strict evaluation scheme and 16.9% F-score in the relaxed evaluation scheme. The rule-based system performs best among all the models. It shows the overall performance improvement of 18.8% and 19.2% F-scores over the *baseline* model in the strict and relaxed evaluation scheme, respectively. The rule-based system also demonstrates the superior performance compared to the CRF based system with the overall performance improvement of 2.1% and 2.3% F-scores in the strict and relaxed evaluation schemes, respectively.

We observe that the rule-based approach is able to correctly identify more than 77% of the unknown event time relation at about less than 23% error rate. The one possible explanation behind the inferior performance of the CRF based system is that we do not have enough instances of all the six types relations, namely BEFORE, BEFORE-OR-OVERLAP, OVERLAP, OVERLAP-OR-AFTER, AFTER or VAGUE, in the available training data. In Task B training data, there are 55% examples of BEFORE relation, 25% examples of OVERLAP relation, 15% examples of AFTER relation and the rest 5% examples are for other relations. Rules can easily capture the properties of three most occurring relations in the training data. The CRF based system performs better with the inclusion of tense feature. This tense feature also plays a crucial role to identify the BEFORE, AFTER and OVERLAP relations under the rule-based framework. Moreover, rules can also easily capture the tense (past, present, infinitive etc.), aspect (perfective, progressive etc.) and POS (noun, verb, adjective etc.) features. But, it is to be noted that in the present experimental set up, we trained CRF only with the context, aspect, POS and tense features.

Table 3. Overall evaluation results

Technique	Strict			Relaxed		
	precision	recall	F-score	precision	recall	F-score
Baseline	57.1 %	57.1 %	57.1 %	57.9 %	57.9 %	57.9 %
CRF	74.1 %	73.6 %	73.8 %	75.1 %	74.6 %	74.8 %
Rule-based	75.9 %	75.9 %	75.9 %	77.1 %	77.1 %	77.1 %

5 Conclusion and Future Work

In this paper, we have reported our work on temporal relation identification that involves the identification of event-time relations in the TempEval-2007 evaluation exercise. We developed two systems, one using CRF and the other using rules, for solving the problem of Task B that involves identifying the six different relations between the events and the document creation time. The CRF based system was developed using some of the features available in the TimeBank corpus. The rule-based system has been developed by identifying a set of handcrafted rules by investigating the various features in the training dataset. Evaluation results show that the rule-based system performs better than the CRF based system.

We would like to experiment by incorporating all the features available in the training data in to the CRF framework. In future, we also want to introduce additional features that may be extracted from our existing tools. Some rules may be identified to make the system more robust. Future works also include investigating other statistical learning techniques like Maximum Entropy and Support Vector Machine for solving the problem.

References

1. Verhagen, M., Gaizauskas, R., Schilder, F., Katz, M.H.G., Pustejovsky, J.: SemEval-2007 Task 15: TempEval Temporal Relation Identification. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Prague, pp. 75–80 (2007)
2. Boguraev, B., Ando, R.K.: TimeMLCompliant Text Analysis for Temporal Reasoning. In: Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), pp. 997–1003 (2005)

3. Mani, I., Wellner, B., Verhagen, M., Lee, C.M., Pustejovsky, J.: Machine Learning of Temporal Relation. In: Proceedings of the 44th Annual meeting of the Association for Computational Linguistics, Australia (2006)
4. Chambers, N., Wang, S., Jurafsky, D.: Classifying Temporal Relations between Events. In: Proceedings of the ACL 2007 Demo and Poster Sessions, Prague, Czech Republic, pp. 173–176 (2007)
5. Mani, I., Wellner, B., Verhagen, M., Pustejovsky, J.: Three Approaches to Learning TLINKs in TimeML. In: Technical Report CS-07-268, Computer Science Department, Brandeis University, USA (2007)
6. Min, C., Srikanth, M., Fowler, A.: LCC-TE: A Hybrid Approach to Temporal Relation Identification in News Text. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Prague, pp. 219–222 (2007)
7. Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer, A., Katz, G., Radev, D.: TimeML: Robust Specification of Event and Temporal Expressions in Text. In: Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5), Tilburg (2003)
8. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML, pp. 282–289 (2001)
9. Sha, F., Pereira, F.: Shallow Parsing with Conditional Random Fields. In: Proceedings of NAACL 2003, Canada, pp. 134–141 (2003)
10. Pustejovsky, J., Hanks, P., SaurI, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The TIMEBANK Corpus. In: Proceedings of Corpus Linguistics, Lancaster, pp. 647–656 (2003)

Question Answering for Not Yet Semantic Web

Miloslav Konopík and Ondřej Rohlík

University of West Bohemia, Department of Computer Science,
Univerzitní 22, CZ-306 14 Plzeň, Czech Republic
konopik@kiv.zcu.cz, rohlik@kiv.zcu.cz

Abstract. In this paper we present a prototype implementation of the question answering system for one of the inflectional languages – Czech. The presented open domain system is especially effective in answering factual wh-questions about people, dates, names and locations. The answer is constructed on-the-fly from data gathered from the Internet, public ontologies, knowledge of the Czech language, and extensible template system. The system is capable of semiautomatic learning of new templates as well as both statistical and semantic processing of Internet content.

Keywords: question answering, information extraction, machine learning, NLP, morphology, semantic compatibility.

1 Introduction

Question answering (QA) engine is a computer system able to retrieve answers to questions posed in natural language.

Traditionally the QA is expected to provide one concise answer to the user's query. For question *"When did Thomas Jefferson die?"* the ideal answer might be *"July 4, 1826"* with *"Thomas Jefferson died on the Fourth of July, 1826"* being just a little less than optimal.

QA provides user-friendly experience to desktop computer users saving their time by elimination of the hassle to skim through web pages returned by traditional search engine to dig out the sought information. For cell phones or PDA users fast navigation through several web pages as offered by a search engine is even more difficult making single and directed answer even more attractive.

2 Answer to the Challenge of the Semantic Web

The original paper [1] described an expected evolution of the existing Web to a Semantic Web in which the meaning (semantics) of information and services on the web is defined, making it possible for the web to "understand" and satisfy the requests (of both people and machines) to use the web content.

This would make the vast amount of information available in the Internet an ideal source of data for any QA system. At the time the Semantic Web will come true all sorts of advanced syntactic, semantic and contextual processing can be used to answer the question including temporal and spatial reasoning, named-entity recognition, relation

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 125–132, 2010.

detection, coreference resolution, syntactic alternations, word sense disambiguation, logical inferences and commonsense reasoning. All these techniques, in order to be utilized to its full potential, requires the Semantic Web enabling technologies to be extensively used by the publishers of new content (while there also must be a way to "update" the old content and turn it into Semantic Web compliant form). However, the above discussed evolution has yet to occur. Recently number of Semantic Web enabling resources, services, and tools has become available, but still the idea remains largely unrealized [2].

While the slow adoption of the Semantic Web technologies is likely to hinder the exploitation of the Semantic Web benefits, we propose a QA less reliant on the semantics but still capitalize on the large volumes of redundant data and uses NLP techniques to compensate for missing full-fledged semantic analysis.

In this paper we present a generic QA system that can answer questions in Czech. Most importantly the presented system is not tight to any particular domain, database or web service; hence it can answer questions on virtually any topic. In practice, however, we limit ourselves to questions to factual information which are the most common for the scenarios envisaged at the beginning of the paper. We deliberately ignore the semantic-heavy and/or database-based solutions because we see both of them as complementary to our solution.

3 State of the Art

Today the QA is an established part of information retrieval (IR) combining computer science, linguistics, and statistics while it requires much more complex natural language processing techniques than other types of IR systems such as document retrieval. Natural language search engines are thus often regarded as the next step beyond search engines.

Last years research interest moves away from closed-domain systems tailored to a specific domains, which thanks to the possibility to use specific well-tuned ontologies do not offer enough challenge, to open-domain systems that can only relay on general ontologies which are very popular today.

In terms of methods and techniques the systems are divided by the level of understanding of the question to shallow and deep.

Shallow methods are based on keyword search with the aim to locate interesting snippets of text and later filtered and rank such candidate text using syntactic features such as word order or similarity to question.

Due to their syntactic nature shallow systems are typically based on templates. This does not necessarily mean that the templates are trivial and that their definition needs human intervention. One of the recent TREC competition[1] winners capitalized on a simple yet clever algorithm [3] to learn patterns automatically from the Web using very minimalistic training data.

Deep methods on the contrary use more sophisticated syntactic, semantic, and contextual processing to extract and construct the answer. Deep methods have arguably stronger theoretical background but suffer from small knowledge bases needed to

[1] The Question Answering Track of the Text REtrievel Conference (TREC) last ran in 2007.

work effectively. So far, commercially available QA services as well as state-of-the-art research prototypes have avoided Web content due to its unreliability which makes is unusable for reasoning.

The above referenced systems are designed for QA in English. There is only one QA system for Czech language: UIO[2]. The system was developed between 2003 and 2006 and is a multi-domain shallow closed-domain QA system [4].

4 System Structure

The structure of the question answering system is shown in Fig. 1. System functions can be decomposed into several functional blocks.

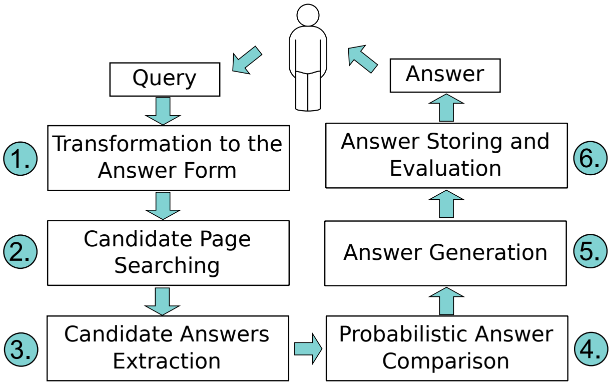

Query
Answer
1.
Transformation to the Answer Form
2.
Candidate Page Searching
3.
Candidate Answers Extraction
4.
Probabilistic Answer Comparison
5.
Answer Generation
6.
Answer Storing and Evaluation

Fig. 1. The schema of our question answering system

1. A query from a user is at first transformed to expected answer forms. The transformation is done according to the selected template. The appropriate template is determined from the user query using template matching algorithm. For example if a user asks the system: “When did Charles IV die?”, the transformation results may be the following: “Charles IV died”, “Charles IV (-)” etc. We call the result of the transformation *the answer form*.
2. Next, each answer form is sent to an ordinary keyword-based Internet search engine (e.g. Google^{TM}). The engine returns pages that contain words from the answer forms. First N (in our case 100) pages are downloaded as *candidate pages* and further processed in the system.
3. In the next step the template is used again – now to determine what type of an answer the system should be looking for. In the above example query the system should be looking for a date. The candidate pages that contain the right type of the answer are passed to next step of processing as *the candidate answers*.

[2] From Czech Umělá inteligence opice – Monkey Artificial Intelligence.

4. Next, the candidate answers are the subject of statistical processing. The goal is to find out the right (factually correct) answer among all the candidate answers. The basic idea is that the answer, that is the most frequent, is the correct one. However to do the statistical exercise right it is necessary to know which answers are semantically equal and which are not. For example if we consider date, the following phrases are compatible: “February 1328”, “1328”, “1.2.1328” and the following phrases are not: “1922”, “14th century”. Obviously, such relation of semantic compatibility is not symmetric. More precise facts are scored higher than compatible yet less precise facts.
5. The next step is answer generation. The system provides two types of answers – concise and contextual. In the above example the first one is the date of birth. The second one is an answer containing the found date in the form of an entire sentence. In order to retain grammatical correctness the entire sentence it is not generated but rather extracted from the candidate page. The returned sentence is the shortest found sentence containing the date.
6. Lastly, the found answer is presented to the user. At this point the user is also asked for evaluation. If the user selects that the answer is correct or incorrect such evaluation is stored to be used for further optimization of templates (see section 5).

5 Templates

Templates are used in the system for two basic purposes:

1. for transformation of the query to the phrase that is sought in the Internet,
2. for extraction of the sought information from the candidate pages.

Templates consist of two basic parts: *parameterized query* *parameterized search pattern*.

Table 1. Token types in templates

Token Type	Explanation
word	a word form
lemma	a lemmatized word
POS tag	a POS tag of a word
named entity (NE)	a named entity, there are 6 types of NEs defined in the system: DATE, TIME, DATE&TIME, NUMBER, PERSON (a name of a person), GEOGRAPHY (a geographical proper name)
semantic category	a semantic category in an ontology, currently we use SYNONYMs and HYPERNYMs, the source of the ontology relations is the Czech Wordnet [5].

The *parameterized query* is used for query identification. It is used to determine what type of the query there is in the input before the query can be transformed for searching. The parameterized query consists of tokens. The types of tokens are listed in Table 1.

The word type is the basic type of a token. Such a token has to be matched exactly as it is written. Other types are a generalization of words. For example the semantic category SYNONYM of word “die” matches all synonymic verbs like “pass”, “perish”, “exit”, “go” etc.

The *parameterized search pattern* is used to search for the answer. It consists of the same token types as the query except the tokens can be labelled. The first purpose of labelling is to know whether there is a relation between the token from the parameterized query and a token from the parameterized search pattern. The second purpose of labelling is to mark the token that is the sought answer. The structure of the template may be more obvious from the following XML code snippet where the first pattern matches sequences like “Charles IV passed in 1378 in Prague” while the second pattern matches “Charles IV (1346 – 1378) was the king of Bohemia”.

```
<template id="1">
    <query> <token id="1">When</token>
            <token id="2" mod="OPT">did</token>
            <token id="3" type="PERSON"/> <!-- ref="3" points here -->
            <token id="4" type="SYNONYM">die</token>
    </query>
                                        <pattern id="2">
    <pattern id="1">                        <token id="1" ref="3"/>
        <token id="1" ref="3"/>             <token id="2">(</token>
        <token id="2" ref="4"/>             <token id="3" type="DATE"/>
        <token id="3" mod="ANSWER"/>        <token id="4">-</token>
    </pattern>                              <token id="5" mod="ANSWER"/>
                                            <token id="6">)</token>
                                        </pattern>
</template>
```

The described template system is capable of semiautomatic learning. The parameterized query has to be always created by hand however the parameterized search pattern can be learned. Before the learning the system is provided with pairs of an input question and a correct answer. For example: “When did Charles IV die?” and “1.2.1328”. The system then starts a batch of searches. Some words like “When” and “?” are on the stop list and they are not used for learning. Synonyms of other words are generated using the *semantic compatibility relation* (see Section 6). All the words and their synonyms are then searched on the Internet. A modified vector model method is used for searching the Web. It is required that at least one of the correct answer synonym has to be present in the result. After this step the *learning corpus* is available. In the learning corpus the correct answer synonyms are found and the sentences that contain them may be examined. These sentences are used as the base of templates. At the beginning the sentence itself is used as the parameterized search pattern. The new pattern is used for searching of other known facts (e.g. Václav III and 4.8.1306,. . .). Further, the algorithm iteratively replaces the words from the new pattern with more general tokens until the following criterion is optimized: *Perf* $\frac{P \quad N}{T}$ where P is the number of positive matches, N is the number of negative matches and T is the total number of performed searches.

6 Looking for the Answer

This section contains an explanation how is the answer being found using templates and the *semantic compatibility relation*. In the first step the parameterized search patterns are used to extract candidate answers from candidate pages. The sentences that match a parameterized search pattern make the candidate answers.

The second step is to extract the correct answer form candidate answers. The answers are statistically processed. To do so a statistics of the answers is created. It is counted how many times an answer occurred in the candidate answer set. The answer that was the most frequent is presumably the correct one and it is the result.

In order to make the statistics more accurate the semantic compatibility relation was defined. In principle, a piece of information is compatible with another one if the first piece of information does not contradict (any value in) the second one. In the following we present several special instances of this relation.

Calendar Date and Time
A date is compatible with another date if and only if none of the values from the first date contradict any value from the second date. For example if the first date specifies only a year and the second date has a day-month-year format then the first date is compatible with the second one if the years are equal. The second date in this case cannot be compatible with the first one because it is more specific. The same principle is valid for time information and combination of date and time information.

Of course, the date and time has to be normalized before comparison. The normalization means that we transform all ways of expressing a date or a time into one unified format (e.g. phrase "1. February 1328" is transformed to "1.2.1328").

Number
The principle for number compatibility is very similar to calendar date and time principle. A number is compatible with another number if and only if the integer and fractional parts are equal or the fractional part in the first number is not specified. We also use approximate numbers and rounding (e.g. "Prague has over 1 million inhabitants."). Numbers are normalized prior processing. Normalization includes handling of "." or "," to delimit integral and fractional part of numbers as well understanding numbers written in words.

Person names
The principle is again similar to previously mentioned cases. Person names are split to the title before name, first name, middle names, surname, number (in case of kings) and title after name. Any part can be shortened to its first letter (e.g. Johann Sebastian Bach to J. S. Bach). Similar to dates a shortened name may be compatible with the full name, but not vice versa.

Geographical names
The compatibility relation for geographical names is handled differently. There is no analytical approach similar to previous cases. We manually created a relationship database for the most frequent names. Despite the different nature of the relation it is also not symmetric (e.g. America is compatible with USA, but not vice versa).

All the aforementioned relations have the generative capability. It means that given a phrase the compatible phrases can be generated (e.g. given the date "1. 2. 1328" the following phrases are generated: “February 1328”, “1328” etc). This property of the relation is used for template learning (see Section 5).

To simplify the explanation we have so far omitted one type of questions. It is the kind of questions that begin with words like “Why”, “How”, “What” and other similar questions. The answer to such question is rarely a single sentence, instead it is usually rather long explanation comprising several sentences. For example the question “Why is the sky blue?” is answered as a short paragraph consisting of 4 or 5 sentences.

In our template system the summarization is described by a specialized parameterized search pattern. The pattern contains a token that has the type called “SUMMARIZE”. The rest of the pattern is used for query transformation. The parameterized query itself is in no way different from other types of questions.

If there is a token called “SUMMARIZE” in the query a simple summarization is performed. It is based on a simple statistics. First five sentences that start with the words from the transformed answer are examined for each page in the candidate page set. The unigram word model is build from the examined sentences. The five sentences that obtain the highest probability in the model are returned as the answer. In this way the answer that contains the most similar words with all the other answers is returned. Obviously, this is not an optimal solution but it works surprisingly well.

7 Results and Future Work

There is currently no established and objective testing method for QA systems. We have prepared a set of 100 testing questions and evaluated our system on them. We achieved 64% of correct answers, 7% of partial correct answers (the answer was correct but the presentation of the answer was not natural) and 29% of incorrect answers. Of course, these results must be considered only illustrative. We are aware that they are neither objective nor comparative.

In the following we dicuss our future work which is focused on two areas: (1) effective identification and dealing with enumerative named entities and (2) employing rigorous summarization methods for answer generation.

Currently we distinguish two types of named entities in our system: analytical (date, time, number) and enumerative (person and geographical names). The *analytical named entities* are described by grammars and regular expressions. We use the so called semantic active tags mechanism [6] for semantics extraction and the semantic compatibility of these NEs. Active tags provide a mechanism to obtain a good coverage of all possible analytical NEs.

To achieve a good coverage of the *enumerative named entities* is much harder task. We have downloaded several Internet databases (e.g. Wikipedia, national registers and other freely available lists) to obtain a solid database of enumerative NEs. However, to obtain the compatibility relation remains to be a problem. There is no database on the Internet that says that sometimes phrases like “America” or “states” are used in the same meaning as the phrase “USA”. Currently, we are exploring the possibility to use latent semantic analysis (LSA). LSA method automatically finds semantically related words. Obviously, the LSA is not a silver bullet. We will have to explore methods to determine not only the type of the found semantic relation but also methods to automatically determine the threshold to say whether the two expressions are semantically related or not.

In the section 6 we have explained a simple *summarization method* that is currently being used in our system. Despite being efficient we are not comfortable with its ad-hoc nature. Our future effort is focused on using better summarization methods. So far we failed to use any state-of-the-art summarization method directly. Due to rather unusual nature of our task some modifications of the existing methods are required.

8 Conclusions

In this article we have presented a design of a question answering system. The proposed system is meant the be an extension of a traditional keyword-based search engine. Our system is not very effective for querying dynamic databases (like shopping databases, public transport databases etc) but to such systems it is meant to be complementary rather than competitive.

The advantage of the system is that it can share resources with an ordinary indexing search engine (the main database of indexed pages is identical for both systems) which makes it a low-cost solution. The system is based upon reliable methods (templates, index search etc.) that are extended with a simple yet effective statistics. It results in a system that returns surprisingly accurate answers for a wide range of questions. To test our system you can access *removed for a blind review*. The main contribution of our design lays in the mechanism of template learning and the definition of compatibility relation. Effective use of the natural language processing methods makes is possible to offer an open-domain QA system for highly inflectional language such as Czech. After implementation of future improvements (see section 7) we expect a further increase of an already solid answer precision and user experience.

Acknowledgment

This work was supported by grant no. 2C06009 Cot-Sewing. We would like to thank the students Tárik et al. for implementation of a significant portion of the system.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Sci. Am. Mag. (2001)
2. Shadbolt, N., Hall, W., Berners-Lee, T.: The Semantic Web Revisited. IEEE Intelligent Systems (2006)
3. Ravichandran, D., Hovy, E.: Learning Surface Text Patterns for a Question Answering System. In: Proceedings of the 40th ACL, pp. 41–47. ACL, Philadelphia (2002)
4. Svoboda, L., Popelínský, L.: Communication with WWW in Czech. Kybernetika 40 (2004)
5. Pala, K., Smrž, P.: Building Czech Wordnet. Romanian Journal of Information Science and Technology 2004 (7), 79–88 (2004)
6. Habernal, I., Konopík, M.: Active Tags for Semantic Analysis. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 69–76. Springer, Heidelberg (2008) ISBN 978-3-540-87390-7

Automatic Acquisition of Wordnet Relations by Distributionally Supported Morphological Patterns Extracted from Polish Corpora

Roman Kurc[1], Maciej Piasecki[1], and Stan Szpakowicz[2,3]

[1] Institute of Informatics, Wrocław University of Technology, Poland
{roman.kurc,maciej.piasecki}@pwr.wroc.pl
[2] SITE, University of Ottawa, Canada
szpak@site.uottawa.ca
[3] Institute of Computer Science, Polish Academy of Sciences, Poland

Abstract. *Espresso* is a pattern-based algorithm of extracting lexical-semantic relations, defined for English. We present its adaptation to Polish. We consider not only the technicalities such as the availability of language-processing tools for Polish, but also pattern structures which leverage the specificity of a strongly inflected language. We propose a new method of computing the reliability measure of extraction; this leads to a modified algorithm which we have named *Estratto*. In this paper we investigate the influence of additional lexico-semantic data and information from generic patterns.

Keywords: lexical-semantic relations, pattern-based relation extraction, *Espresso*, *Estratto*, wordnet expansion.

1 Introduction

Pantel [1] names two paradigms for relation extraction: based on *patterns* and on *clustering*. The latter has good recall but problematic precision, because its typical product is a generic *measure of semantic relatedness*, without pointing to specific lexical-semantic relations. The clustering-based paradigm has been studied for Polish [2,3], but there is little on pattern-based extraction. Pattern-based approaches have better precision, but very low recall if patterns are constructed manually [4]. To increase recall, more – or more generic – patterns are required. Extracted automatically from a corpus, they tend to have broad coverage but intrinsically low precision. In contrast with clustering-based methods, however, patterns allow the extraction of instances of a particular relation. Pantel and Pennacchiotti [1] present a very successful pattern-based method, *Espresso*.

Espresso's success has motivated us to adapt its approach to Polish. We seek statistical extraction of lexico-morphosyntactic patterns for the purpose of semi-automatic acquisition of hypernymy in the *plWordNet* project [5]. The idea is to assist the linguists' manual work by suggesting lexical units for addition. Higher recall is thus desirable, even at the expense of precision. We first adapted the *Espresso* algorithm to the Polish language and to the reality of a limited availability of language processing

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 133–141, 2010.

tools for Polish. That experience has led to an extended version of *Espresso*, which we named *Estratto*. *Estratto* surpassed *Espresso*, but many challenges remained. We present the effects of that further study and development of *Estratto*.

First, we want to improve the pattern extraction mechanism towards better coverage of more specific, higher-precision but less frequent patterns. Second, instance evaluation is recursively based on the previous pattern evaluation. The whole process can migrate from extraction of instances of certain *preselected* lexico-semantic relation (or relations) to the extractions of word pairs with less well defined semantic associations. For the purpose of instance evaluation, therefore, we need a measure of semantic relatedness extracted from the corpus as a second knowledge source which helps focus the algorithm on a handful of lexico-semantic relations. Experiments show that this approach allows the extraction of more instances, with no detriment to the precision of results.

2 Basic Extraction Algorithm

The *Estratto* algorithm, just like *Espresso*, has three phases: pattern *construction* from instances of a relation, statistical *evaluation* of the patterns, and *extraction* and evaluation of instances using patterns marked as positive.

The first phase extracts both generic and specific patterns – more on that in Section 4. The last phase uses those patterns to find instances. Having generic patterns usually lowers accuracy but improves recall. This balance is achieved by the proposed measure of *reliability* of patterns and instances, explained below. Patterns are evaluated and ranked by a measure of reliability, whereas instances are evaluated by a measure of *confidence* based on reliability. Only the best patterns and instances are kept for the following phase of the algorithm.

A pattern's reliability originates from its strength of association with reliable instances and the number of reliable instances which extract it [1]:

$$r_{\pi}(p) = \frac{\sum_{i \in I} (pmi(i,p) * r_t(i)) * d(I,p)}{\max_P \left(\sum_{i \in I} (pmi(i,p) * r_t(i))\right) * |I|} \tag{1}$$

p is a pattern, i – an instance, r_t – a reliability measure for instances; pmi is Pointwise Mutual Information [1]; I is the set of instances, $|I|$ – its cardinality; $d(I, p)$ counts unique instances with which pattern p is associated.

PMI tends to produce significantly higher values for combinations of lower-frequency patterns and instances (e.g., fewer than 10) [1]. Following [1], we compensate by multiplying PMI values by a *discounting factor* proposed in [6].

The reliability of instances is defined very similarly (P is the set of patterns).

$$r_{\pi}(i) = \frac{\sum_{p \in P} (pmi(p,i) * r_t(p)) * d(P,i)}{\max_P \left(\sum_{p \in P} (pmi(p,i) * r_t(p))\right) * |P|} \tag{2}$$

The measure of confidence of an instance extracted by generic patterns is based on the application of specific patterns of high reliability:

$$S(i) = \sum_{p \in P_R} S_P(i) * \frac{r_{\pi}(p)}{T} \tag{3}$$

P_R is the set of specific patterns, $S_p = pmi(i, p)$ and T is the sum over the reliability of specific patterns.

Except at the first iteration of the *Estratto* algorithm, both types of reliability are calculated from the corpus frequency of the best instances extracted during the preceding iteration. The algorithm has limited precision (see the baseline experiment – the unmodified *Estratto* – in Section 6), so the data for the reliability calculation include incorrect instances which represent different forms and grades of semantic association. The subsequent iterations may extract instances which drift away from the area of the preselected lexico-semantic relations.

We have thus far avoided a drift in *Espresso* and *Estratto* by filtering instances for the next iteration according to a predefined confidence threshold. Its value must be set appropriately – this is crucial to the algorithm's stability in terms of the relations represented by the iteratively extracted instances. Here, we want to analyse experimentally how well a measure of semantic relatedness serves as an additional criterion in instance selection. Given a pair of lemmas, the measure returns a value from the range $\langle 0, 1\rangle$. It is constructed automatically using the data extracted from a corpus on the basis of on the *Rank Weight Function* (RWF) algorithm [2,5]. The measure is based on clustering – fundamentally different from *Estratto*'s pattern-based nature – so it can be treated as a knowledge source external to the *Estratto* algorithm.

In one of the experiments, we replaced PMI values in (1) with values of the RWF measure to be used directly as the instance reliability values. In this setting, instance reliability does not depend on history, but only on an independently constructed measure. Section 6 also presents an application of the RWF-based measure as a direct tool for the selection of instances for the next iteration.

3 Patterns

Espresso was developed for English. In freer-order languages, like Polish, inflection often replaces or supplements position as a source of information about structure. There is elaborate morpho-syntactic agreement to account for. Given that the largest available Polish corpus is the IPI PAN Corpus (IPIC) [7], we adopted its formalism: grammatical classes (more fine-grained than the traditional parts of speech) and categories (case, number, gender etc.). The position of a lexical unit (LU) in a sentence need not be correlated with its role in an *asymmetrical* lexical-semantic relation. For example, most patterns assign different case values to the two arguments in hypernymy: hyponym (subclass) and hypernym (superclass). While we could generate specific patterns for all different combinations, it is better to look for a generalisation of a group of patterns.

Multiword LU (MLUs) are best identified using either chunking, or patterns with collocations. Chunking or shallow parsing requires the appropriate tools – something not yet available for Polish as a plug-in. In *Estratto*, we are particularly interested in *rare* instances, so using collocations alone is not an option. We adopt a straightforward manual solution. We perform corpus search for collocations and filter them manually. Next, we add MLUs from dictionaries to the collocation list. *Estratto* uses two types of MLUs, noun-noun pairs and adjective-noun pairs, and considers the MLUs' relative position (fixed or flexible) and agreement between constituent words. The resulting list of MLU is then used during the process of acquisition of patterns and instances.

Following [4], patterns are regular expressions in which the terminal symbols are lemmatized word forms, and variables correspond to noun phrases. We assume regular expressions with Kleene closure but without grouping, and morphological base forms of LUs as terminal symbols. Patterns are flat, and describe a sentence as a sequence of word forms or small groups of word forms. No deeper syntactic structure is involved. There are three types of terminal symbols: the wild-card symbol $*$, a *base form* and a *matching location*. The base form is a lemma of some LU together with its *grammatical class*.

Part of a pattern which signals a *matching location* encodes certain expected values of morpho-syntactic parameters of matching LUs. The grammatical classes in IPIC are very fine-grained. We regroup them, so for example noun represents all of *substantive*, *gerund*, *foreign nominal* and *depreciative noun*. A matching location is a reduced version of IPIC's morphosyntactic tag, in which only some grammatical categories are specified (see Table 3). As in [1], there is always a matching location at the beginning and at the end of a pattern.

4 Induction of Patterns and Extraction of Instances

Pantel [1] writes that patterns can be inferred by any pattern-learning algorithm. In *Espresso*, the inferred patterns are then generalized: *terminological labels* replace all *terminological expressions* (noun phrases). The unavailability of a Polish chunker working with high precision for any text precludes such generalization, so we chose a slightly different method. Extracted patterns are grouped and then merged [8]. Generalization begins by adding a wild-card right after the first matching location and just before the second. Patterns thus extended are then clustered. Each cluster contains patterns with the same morphology (matching locations) and similar sequence of words (lemmatised and described morphologically) located between the matching locations. The patterns in clusters are then merged using the longest common substring algorithm. The algorithm is guided by a predefined list of relation-specific lemmas or individual word forms (about 20 in total). For hypernymy we have, among others, 'być' *to be*, 'stać się' *to become*, 'taki' *such*, 'inny' *other*. Next, we compact the patterns. Both general and specific patterns are clustered by their lexical structure. The extraction of relation instances (pairs of LUs) comes after pattern induction and selection.

A generalized pattern is classified as *generic* either if it was created by *generalising* a group of specific patterns, or if the number of instances it extracted is *significantly large* in comparison with specific patterns. Our interpretation of *significantly large* varies from the rule '*frequency n times higher than the most frequent specific pattern*' [1] to '*more frequent than any pattern on average*'.

We noted that some specific patterns can extract as many instances as generic patterns. It was easy to check that such overly productive specific patterns yield instances of lower quality than regular specific patterns. It was thus reasonable to treat those specific patterns and generic patterns similarly. So, we experimented with both conditions mentioned above; see Section 6 for details.

Concurrently, we experimented with combining information from generic and specific patterns. We introduced an additional *pattern combination heuristic* meant to modify the role of patterns which have been classified as generic: select an instance for the next iteration only if it is covered by at least one specific patterns. This heuristic stems from the observation that specific patterns are very likely to extract very good instances. If, however, few instances are extracted, it is very difficult to determine the heuristic's reliability. That is why generic patterns help validate the quality of extracted instances in terms of statistical evidence. The proposed heuristic seems to gain additional support for specific patterns and utilise the generic ones without introducing many noisy instances.

We also applied the RWF measure of semantic relatedness (Section 2) to direct filtering of instances. In some experiments, we accepted for the next iteration only instances present on the lists of the *n* lemmas most semantically related to the given one. We will refer to this technique as a *filter of the n most related*. We assume, similarly to [9], that a generic pattern introduces a systematic error. The error can be intuitively understood as extracting semantically unrelated lemmas only because of their close co-occurrence in text. The proposed filter causes the extraction only of instances related by a similar distribution in the whole corpus.

Several parameters control pattern induction and instance extraction. They were set according to the results obtained in [8].

5 Performance Measures

It is generally difficult to evaluate properly the effects of the extraction of lexical-semantic relations [10,11]. There are two ways of judging instances: find them in an existing manually constructed resource – *plWordNet* in our case – or rely on human judgment. The former introduces a bias so long as *plWordNet* is not large enough. We decided, therefore, to evaluate manually representative samples of the extracted lists. We used two evaluation measures.

1. *Precision based on human judgment* is evaluated according to a randomly drawn sample from the list of instances. This evaluation measure was used only for the first group of experiments (Section 6). The error level of the sample was 3% and the confidence level was 95%. During evaluation, each instance goes into one of the predefined classes, described below:
   (a) *P* – proper linguistic hypernymy, as in wordnets;
   (b) *PT* – a form of conceptual hypernymy supported by local context;[1]
   (c) *PG* – instances which would be correct given sufficiently smart linguistic processing tools;[2]
   (d) *hypo* – instances already added to *plWordNet*;
   (e) *F* – a “catchall” class (that is an error given *Estratto*'s goal).
   We will denote the combination **PT**+P+PG+hypo as *allHypo* and P+PG+hypo as *lingHypo*.
2. *Relative recall* is a measure based on a proportion of the number of instances extracted in the given experiment and extracted by the baseline algorithm.

[1] Examples: a relation linking a named entity with its hypernym signalled by the head noun; or a single-word LU as a remote hypernym in place of a multi-word LU.

[2] For example: wrong number (Carpathian Mountains versus mountain) or wrong – but semantically related – lemma (tournament versus compete).

6 Experiments

All experiments were performed on IPIC [7] ($\approx$ 254 million tokens). The corpus is not balanced, but covers several genres. Our experiments focus on the two major extensions to the previous version of *Estratto*, proposed in this paper:

- the modified way in which the information gained from generic patterns is used in the extraction process – the pattern combination heuristic,
- and the introduction of additional knowledge based on distributional semantics delivered to the algorithm.

To meet these goals, we tested the following configurations of *Estratto*.

a) **EST** *Estratto* with generic patterns, exploiting specific features of Polish language and the extended reliability measure (Section 4). The results achieved in this configuration became a point of reference – a baseline.

b) **EST+sp** The baseline algorithm plus the pattern combination heuristic (Section 4), according to which instances extracted by the generic patterns must be supported also by at least one specific pattern; only patterns created as a generalisation of specific ones were treated as generic.

c) **EST+sp+avg** Similar to (b), but patterns are classified as generic only on the basis of their frequency (see Section 4 and ways of identifying generic patterns defined there);

d) **EST-dir** The baseline extended with yet another filter: the removal of patterns which tend to extract reversed instances – in a valid but reversed semantic relation (e.g., hypernymy when hyponymy is being extracted).[3]

e) **EST+flt** The baseline plus the filter of the *n* most related instances (Section 4); it keeps instances supported by the lists of the *n* most semantically related lemmas produced using RWF.

f) **EST+RWF** The baseline in which PMI was exchanged in calculating reliability with the measure of semantic relatedness based on RWF (Section 2).

If not stated otherwise, the threshold instance confidence for all EST configurations is 1.5 (established experimentally). The number of *top k* patterns increases by 1 in every iteration. At the beginning there were 8 patterns [8]. There were 8 cycles. All experiments presented here were performed on IPIC and were focused on the hyponymy/hypernymy relation.

In the first group of experiments we analysed the effect of modified treatment of generic patterns – configurations a), b), c). See Table 1. It is clear that restricting generic patterns increases precision and decreases recall. On the other hand, recall is still higher in comparison to results obtained using only specific patterns. Comparing configurations b) and c) shows that of 1,109 filtered instances only 25% correct instances are removed. So, detection of overly productive specific patterns additionally

[3] This filter was motivated by the observation that some pattern types (e.g., patterns based on coordinate NPs with certain conjunctions) may be induced and at the same time used as patterns producing instances of the inverse relation. Given a pair of mutually inverse patterns, the filter removes the pattern with a lower reliability.

Table 1. The number and precission of extracted instances

	#instances total	#allHypo	allHypo	#lingHypo	lingHypo	relative recall of allHypo	relative recall of lingHypo
EST(baseline)	23,044	7,036	31%	4,559	20%	100%	100%
EST+sp	5,831	3,177	54%	2,194	38%	45%	48%
EST+sp+avg	4,722	2,799	59%	1,917	41%	40%	42%
EST-dir	11,755	3,692	31%	2,476	21%	52%	54%
EST+flt	4,137	2,070	50%	1,507	36%	29%	33%
EST+RWF	307	187	61%	168	55%	3%	4%

increases precision. The last observation in this group of experiments is the evident drop between a) and b) of the number of correct instances extracted. This is very interesting, because one can see that there might be a way to increase the number of correct instances by introducing a more elaborate approach to generic patterns.

An additional experiment checked if the previously noted tendency of pattern acquisition to promote contradictory patterns could be minimized by removing one of such patterns. The simple heuristics suggested in d) did not prove useful, so this problem requires further analysis.

Experiments for configurations e) and f) explored the influence of the additional distributional-semantic knowledge. In e) we applied a filter after the phase of instance evaluation. Thus semantically unrelated instances in the ranking were removed. Recall, however, that we only used instances from the lists of 20 lemmas most semantically related by RWF. We expected, then, that recall would be lower. On the other hand there was a visible increase in precision. Thus, distributional data *can* be used in pattern-based approaches to prevent relation extraction from producing instances with only vague semantic connections.

Table 2. Examples of extracted instances

szkoła; instytucja (*school*; *institution*)	maszyna; urządzenie (*machine*; *mechanism*)
kościół; związek wyznaniowy (*church*; *religious association*)	wychowawca; pracownik (*tutor*; *employee*)
kombatant; osoba (*combatant*; *person*)	obligacja; papier wartościowy (*bond*; *security*)
bank; instytucja (*bank*; *institution*)	pociąg; pojazd (*train*; *vehicle*)
telewizja; medium (*television*; *medium*)	prasa; medium (*press*; *mass media*)
szpital; placówka (*hospital*; *establishment*)	czynsz; opłata (*rent*; *payment*)
grunt; nieruchomość (*land*; *real estate*)	Wisła; rzeka (*Vistula*; *river*)
świadectwo; dokument (*diploma*; *document*)	opłata; należność (*payment*; *charge*)
ryba; zwierzę (*fish*; *animal*)	Włochy; kraj (*Italy*; *country*)
jezioro; zbiornik (*lake*; *reservoir*)	jarmark; impreza (*fair*; *entertainment*)
piwo; artykuł (*beer*; *comestible*)	zasiłek; świadczenie (*benefit*; *welfare*)

Configuration f) gave very few extracted instances. There are two reasons. First, it is RWF's nature to signal a relation between LUs or instances even if the function value is very low. This may have decreased reliability. The experiment ran with baseline thresholds, which may have caused excessive trimming of instance list. The other reason is a scarcity of pairs on which RWF was evaluated. In most cases LUs found by Estratto were not on the list.

Table 2 presents examples of instances (hyponym; hypernym) extracted by the *Estratto* algorithm from IPIC. Table 3 shows examples of patterns extracted by *Estratto* from IPIC and used in the extraction of the instances in Table 2.

Table 3. Examples of patterns (English glosses are not part of *Estratto* patterns)

```
occ=31 rel=0.26803 (hypo:subst:nom) być 'is/are' (hyper:subst:inst)
occ=20 rel=0.222222 (hypo:subst:[nom|gen|dat|acc|inst|loc]) i inny
'and other' (hyper:subst:[nom|gen|dat|acc|inst|loc])
occ=26 rel=0.103449 (hypo:subst:[nom|gen|dat|acc|inst|loc]) a inny
'but other' (hyper:base:[nom|gen|dat|acc|inst|loc])
occ=15 rel=0.0684905 (hypo:subst:inst) przypominać 'resemble' (hyper:subst:acc)
occ=41 rel=0.0263854 (hypo:subst:loc) i w inny 'and in other' (hyper:subst:loc)
occ=86 rel=0.00708506 (hypo:subst:nom) stać się 'become' (hyper:subst:inst)
```

7 Conclusions and Further Work

We have discussed two extensions of the *Estratto* algorithm. One of them is based on modifying the importance of generic patterns. The other comes from using supplementary distributional data. Both approaches resulted in better precision, but the number of extracted instances decreased in comparison with the baseline. Recall in the first approach apparently suffers because of the following fact: even if many generic patterns match some instance, the instance will not be accepted unless there is at least one specific pattern associated with it. In case of the second distributional approach, a solution may be obvious. We need to extend the list with RWF results so that the number of resulting instances is bounded not by the size of the list but the values of RWF.

In our recent experiments we also used a mechanism for indicating multiword lexical units (MLUs). Earlier it was sometimes possible to obtain instances consisting of two similar words. For example, given *word office* and *post office*, the resulting instance would be (office, office), because only single words are matched. Now we introduce a manually defined list of MLUs. As a result, we get instances containing these MLUs. The idea has proven useful in the context of further linguistic work on expanding *plWordNet*. Yet, the number of such instances was still low. Therefore we will extend the MLUs list or try to prepare a tool for the detection of MLUs.

Last but not least, the acquired flat of instances cannot be easily imported into *plWordNet*. Such a representation cannot indicate the wordnet classes to which an instance belongs, and the distance between the LUs in the instance. This problem has already been considered in [1] and in [8].

Acknowledgement. Work financed by the Polish Ministry of Education and Science, project N N516 068637 and partially by the Institute of Informatics.

References

1. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proc. 21st COLING and 44th ACL, pp. 113–120. ACL (2006)
2. Piasecki, M., Szpakowicz, S., Broda, B.: Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 99–106. Springer, Heidelberg (2007)
3. Broda, B., Derwojedowa, M., Piasecki, M., Szpakowicz, S.: Corpus-based semantic relatedness for the construction of polish wordnet. In: ELRA (ed.) Proc. Sixth LREC 2008, Marrakech, Morocco. ELDA (May 2008)
4. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceeedings of COLING 1992, Nantes, France, pp. 539–545. ACL (1992)
5. Piasecki, M., Szpakowicz, S., Broda, B.: A Wordnet from the Ground Up. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2009)
6. Pantel, P., Ravichandran, D.: Automatically labeling semantic classes. In: Susan Dumais, D.M., Roukos, S. (eds.) HLT-NAACL 2004: Main Proceedings, Boston, Massachusetts, USA, pp. 321–328. ACL (May 2004)
7. Przepiórkowski, A.: The IPI PAN Corpus: Preliminary version. Institute of Computer Science, Polish Academy of Sciences, Warsaw (2004)
8. Kurc, R., Piasecki, M.: Automatic acquisition of wordnet relations by the morpho-syntactic patterns extracted from the corpora in Polish. In: Proc. of IMCSIT – 3nd Inter. Symp. Advances in Artificial Intelligence and Applications, pp. 181–188 (2008)
9. McIntosh, T., Curran, J.R.: Reducing semantic drift with bagging and distributional similarity. In: Proc. 47th ACL and the 4th Inter. Joint Conf. on Natural Language Processing of the AFNLP, Suntec, Singapore, pp. 396–404. ACL (2009)
10. Zesch, T., Gurevych, I.: Automatically creating datasets for measures of semantic relatedness. In: Proc. Workshop on Linguistic Distances, COLING 2006, Sydney, Australia, pp. 16–24. ACL (July 2006)
11. Piasecki, M., Szpakowicz, S., Broda, B.: Extended similarity test for the evaluation of semantic similarity functions. In: Vetulani, Z. (ed.) Proc. 3rd Language and Technology Conference, Poznań, Wyd. Poznańskie Sp. z o.o., pp. 104–108 (2007)

Study on Named Entity Recognition for Polish Based on Hidden Markov Models*

Michał Marcińczuk and Maciej Piasecki

Institute of Informatics, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, Wrocław, Poland
{michal.marcinczuk,maciej.piaseki}@pwr.wroc.pl

Abstract. Accuracy of a Named Entity Recognition algorithm based on the Hidden Markov Model is investigated. The algorithm was limited to recognition and classification of Named Entities representing persons. The algorithm was tested on two small Polish domain corpora of stock exchange and police reports. Comparison with the base lines algorithms based on the case of the first letter and a gazetteer is presented. The algorithm expressed 62% precision and 93% recall for the domain of the training data. Introduction of the simple hand-written post-processing rules increased precision up to 89%. We discuss also the problem of the method portability. A model of the combined knowledge sources is sketched also as a possible way to overcome the portability problem.

Keywords: named entity recognition, machine learning, Hidden Markov Model, Polish.

1 Introduction

Language expressions referring to extra-linguistic objects are the primary mean of anchoring the interpretation of the text in the reality. This class of expression encompasses Proper Names, but also other types of expressions with definite reference. In Information Extraction those language expressions are traditionally called *Named Entities* (NEs). However, this name is very misleading, as the objects of reference are entities named by the expressions and not *vice versa*.

The identification and classification of NEs (Named Entity Recognition; NER) is a well studied task in the case of English, but only a few approaches have been proposed so far for Polish. Works for other Slavic languages are not numerous, as well. Dominant approaches to NER for Polish are based on the manual construction of rules, heuristics or grammars cf applications in [1,2,3], machine anonymization [4] and translation [5]. Only a few preliminary, alternative works were presented on the application of Machine Learning methods to NER, e.g. Memory Based Learning in [6], Decision Trees C4.5 and Naïve Bayes in [7].

Our general goal is to develop a NER method for Polish based primarily on supervised learning on the basis of manually annotated corpora and supplemented

[*] Work financed by European Union within Innovative Economy Programme project POIG.01.01.02-14-013/09.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 142–149, 2010.

by knowledge expressed manually in form of manually written rules, applied during pre- or post-processing. In the paper we are going to verify the feasibility of the combined approach in a limited setting of the problem, discussed in the following section, and apply a selected Machine Learning method to the problem. Namely, we want to investigate the performance of the model on the basis of *Hidden Markov Model* (henceforth *HMM*) in the Named Entity Recognition task for Polish.

HMM is one of the machine learning methods that have been successfully applied to NER for several languages, e.g. English [8,9], Chinese [10], Dutch and Spanish [11]. The average results are in the range of 60–90% F-measure. According to our best knowledge there were no earlier attempts to apply an HMM in NER for Polish.

2 Task Definition

The exact extension of the NEs differentiates among works from the literature, however there are three main categories of expressions that stay stable as NE subclasses: *persons* (henceforth written *PERSON*), organizations *ORGANIZATION* and *locations LOCATION*. As a machine learning based approach requires prior construction of an annotated training and testing corpus, mostly of a substantial size, we decided to limit the NER task to the recognition of NEs of the *PERSON* subclass for the needs of the initial work presented here.

We assume that NEs of the *PERSON* class are linearly continuous expressions, i.e. each NE is a continuous sequence of words in the text. We assume also that *PERSON* expressions can include subparts representing *first name*, *second name*, *last name*, *maiden name*, *pseudonym* that refer together to an unique person and are not a part of other expressions.

3 Corpora

Concerning the possible future areas of the NER applications we used two available domain corpora as the basis for the preparation of the test and training data, namely: a corpus of stock exchange reports – representing the economy domain, and a corpus of police reports – the public security domain.

3.1 Stock Exchange Reports

The corpus of stock exchange reports consists of 1,215 documents[1] collected from the GPWInfoStrefa[2] published by 185 different companies. The corpus consists of 10,066 sentences, **282,418 tokens** and includes **654 PERSON** annotations.

The reports are written in formal style. Expressions referring to people mostly consist of a first name and a last name and are commonly preceded by a honorific *Pan*/*Pani* ('*Mr./Ms.*'). The text is full of expressions starting from upper character, i.e. (1) aliases (*Company*, *Agreement*, etc.), (2) offices (*Chairman*, *Executive*, etc.), (3) bodies in company structure (*Board of Directors*, etc.) that makes the recognition of the Proper Names more difficult.

[1] http://url.hidden

[2] http://gpwinfostrefa.pl

3.2 Police Reports

The corpus of police reports consists of statements produced by 11 witnesses and suspects. The reports were provided by a local Police Department. Due to the legal reasons the documents were manually anonymized beforehand. The documents were collected within the project on machine anonymization [4]. The corpus consists of 1,583 sentences, **29,569 tokens** and **555 PERSON** annotations.

The documents are written in informal style and contain many one-word person names, mostly pseudonyms and first names. Comparing to the previous corpus, the statements contains a new type of expression "<*person name*> ps. <*pseudonym*>" that is expected to be marked as a one annotation.

The corpus of police reports was used for cross-domain evaluation.

4 Base Line

According to the literature, [4] obtained 93.53% and 88.66% precision in a rule-based anonymizaton of the first and last names, respectively (tested on the 59 Interpol messages). Recall was not measured because the authors could not evaluate the software on original texts, as they contained sensitive data. Another rule-based approach presented by [2] achieved 98% precision and 89% recall (tested on "*about 100 short citations downloaded from Internet or made by testers*"). [1] obtained 90.6% precision and 85.3% recall using manually created grammars and gazetters (tested on 100 financial news articles from the online version of Polish newspaper *Rzeczpospolita*).

A direct comparison with the above results was impossible since we did not have an access to the test data used in those experiments. Thus, we performed two base line experiments described in the following sections.

4.1 Simple Heuristic

The heuristic identifies a continuous sequence of words starting with a upper case character located within limits of the same sentence as *PERSON*. The heuristic obtained only 1% precision (see Table 1) what is caused by two factors: (1) the reports contains many words starting with an uppercase character that are not proper names, and (2) we consider only person names, not all proper names. 42% recall is also below our expectations — many person names are preceded by common words starting with an uppercase character (e.g., *Mr.*, *Chairman*) and are included by the heuristic into annotation. To balance the two factors we prepared a second base line algorithm using a gazetteer.

4.2 Gazetteers

A sequence of tokens is marked as an annotation only if every token in the sequence is found in the gazetteers. We used two gazetteers of the first and last names that included the total number of 63,555 unique entries [1]. The performance obtained with the gazetteers is presented in the Table 1. Comparing to the heuristic, the gazetteers

Table 1. Results of PERSON recognition with the heuristic (HEUR), the gazetteers (GAZE), Hidden Markov Models (HMM) and filtered HMM (+f$_{1+}$ and +f$_{2+}$)

	10-fold CV on the economic domain					Cross-domain		
	HEUR	**GAZE**	**HMM**	**+f$_{1+}$**	**+f$_{2+}$**	**HMM**	**+f$_{1+}$**	**+f$_{2+}$**
Precision	0.85 %	9.47 %	61.82 %	74.49 %	**89.06 %**	28.04 %	**63.61** %	83.49 %
Recall	41.74 %	41.44 %	92.35 %	90.21 %	**89.60 %**	47.74 %	**47.57** %	32.79 %
F$_1$-measure	1.67 %	15.42 %	74.05 %	81.66 %	**89.33 %**	35.33 %	**54.43** %	47.09 %

obtained similar recall of 42% and quite better precision of 9%, however, it is still low. A large number of the False Positives was caused by common words apearing at the beginning of sentences and present in the gazetteers (names that are also common words) and expressions partially matched (i.e. only part of the expression was present in the gazetteers).

5 Hidden Markov Model

5.1 Implementation

In our experiments we used an existing HMM implementation called *LingPipe* [12] that employs first-order hidden Markov models. Following [10] the HMM contains 7 hidden states for every NE type T. Transitions between tags are modelled by a maximum likelihood estimated over training data. Tag emissions are generated by *n*-gram character language models with generalized form of Witten-Bell smoothing. The n defines the maximum length of observed character sequences. The smoothing algorithm has one hyperparameter Θ that modifies the interpolation ratio (higher interpolation ratios favour precision over recall). We used the default *LingPipe* settings, i.e. $\Theta = n = 12$ with *smoothing*. For further description and an example see [13].

5.2 Evaluation

In the evaluation we followed the 10-fold Cross Validation scheme using the corpus of stock exchange reports. The corpus was split randomly on the level of sentences into 10 sets with 66–67 annotations each. The system expressed high recall of 93% but low precision of 63% (see Table 1). To investigate the reason of the low precision we manually checked and categorized every *False Positive*.

5.3 Error Analysis

In the 10-fold Cross Validation we got 373 *False Positives*, which were divided into 5 groups of errors.

- **Incorrect proper name category** — 187 FPs (nearly half) were a full or a partial proper names other then *PERSON*, e.g.: (1) *ORGANIZATION* Polifarb Oliva and (2) *LOCATION* ul. Sienkiewicza *(eng. Sienkiewicza St.)*. The group contains 38 FPs that were correct *ORGANIZATION* and 30 FPs that were correct *LOCATION*. We expect that construction of HMM models for both types will eliminate some of the FPs from this group.

- **Lowercase and non-alphabetic expressions** — 99 expressions violated the structure of *PERSON* annotation defined in Section 2. The group consists of (1) English phrases *(ang.* inside directors*)*, (2) words starting with a lower case character *jest to* najtańsza *oferta (eng. this is the cheapest offer)*, (3) single characters *(696.882 akcje* x *20,1 zł)*, (4) non-alphabetic expressions *** *Inne koszty (eng. *** Other expenses)*. This group of FPs could be easily fixed by applying a simple rule filter.
- **Incorrect annotation boundaries** — 35 partially matched annotations. The common mistakes were: (1) an expression consists of a first and a last name but only first name was matched *Pana* Michała *Nowaka (eng. Mr. Michał Nowak)*, (2) a last name contains maiden name that was not matched *Pani* Marii Nowak*-Kowalskiej (eng. Ms. Maria Nowak-Kowalska)* or (3) an annotation was divided between two sentences due to an error in segmentation Maria M. EOS[3] Nowak. (1) and (2) can be fixed with a rule that extend annotation boundary only if the contiguous token is in a gazetteer. (3) may be fixed after correcting the segmentation or with a rule that will join continuous sequence of annotations into one.
- **Missing annotations** — 18 FPs were missing *PERSON* annotations.
- **Common words starting with an uppercase character** — 6 FPs were common words at the beginning of a sentence.

The analysis of errors showed, that the simplest improvement of the results can be obtained by filtering *lower case and non-alphabetic expressions* with a simple hand-crafted rule.

5.4 Post Filtering

To improve precision of HMM we applied a simple filter that matches only sequences of words starting with an upper case character and optionally separated by a hyphen. The rule was written as the following regular expression:

```
WORD = "([A-Z]|Ą|Ż|Ś|Z̀|Ę|Ć|Ń|Ó|Ł)([a-z]|ą|ż|ś|ź|ę|ć|ń|ó|ł)*";
PATTERN = "/^WORD( WORD)+( - WORD)?( (WORD))?$"
```

The filtering improved precision from 62% to 89% with a very small 1% reduction of recall. The detailed result of filtering is presented in the Table 1. As names in the testing corpus are at least two words long, we assumed the filter matches multiword sequences only. In general, PERSON may consist of only one word (e.g. a last name), so we relaxed the regular expression to match also one-word annotations. As we expected, precision decreased to 74% comparing to the previous filter, but on the other hand, the rule is less baised.

[3] End of Sentence marker.

6 Cross-Domain Evaluation

In order to evaluate the model portability across domains the system was trained on the corpus of stock exchange reports and evaluated on the corpus of police reports. We tested two variants: HMM itself and HMM combined with post filtering. The detailed results for both configurations are presented in Table 1. The HMM-based variant achieved precision of 28% and recall of 48%. Futher filtering of the result improved precision up to 64% while recall stayed unchanged.

The analysis of the errors revealed, that the low recall is mainly caused by too wide matches — the match captures not only the NE but also a word that precedes and/or follows the annotation. Most of the additional words start with a lower case character that may be easily correct with a trimming rule.

Another problem we observed is the incorrect segmentation of structures like "<person name> ps. <pseudonym>" that introduce a pseudonym of a person. In the testing data the expression is marked as a single annotation. In the processing chain the expression is divided into two sentences due to the segmentation errors.

7 Data Coverage and Model Granularity

Within the same domain the coverage of annotation diversity is good due to limited number of names and patterns, and the filtered HMM obtained high result of 82% F-mesaure (even 89% for the baised filter). However, when we consider the coverage on the level of annotation patterns and their compositions the data are insufficient for the general purpose. In [13] we found a list of 9 possible ways of naming PERSON in text while the Stock Exchange corpus covers mostly two-word annotations – 87% of annotations is of the type "<*first name*> <*last name*>" (see Table 2).

Table 2. Coverage of PERSON annotation patterns in the Stock Exchange corpus

	Count	**%**
<*first name*> <*last name*>	579	88.53 %
<*first name*> <*middle name*> <*last name*>	50	7.64 %
<*first name*> <*last name*>-<*maiden name*>	11	1.68 %
<*inital*> <*last name*>	7	1.07 %
<*fist name*> AND <*last name*>	7	1.07 %

The diversity of multi-word annotations can be extended by training the model language on different combinations of atomic elements (*first names*, *last names* etc. in case of PERSON). However, the number of all possible combinations is very large (e.g. 1,089×44,631 combinations of first names and last names on the basis of our gazetteers). We can reduce the number of training instances by changing the model granularity and train the system to recognize atomic elements. To evaluate this approach we have re-annotated the corpus with FIRST_NAME and LAST_NAME annotations (*maiden names* we treated as *last names*). The unbiased filtering produced better results (see Table 3).

Table 3. Result of PERSON recognition with the heuristic (HEUR), the gazetteers (GAZE), Hidden Markov Models (HMM) and filtered HMM (HMM+f)

	PERSON		*FIRST NAME*		*LAST NAME*	
	HMM	**+f$_{1+}$**	**HMM**	**+f$_{1+}$**	**HMM**	**+f$_{1+}$**
Precision	61.82 %	74.49 %	76.05 %	89.13 %	73.73 %	83.06 %
Recall	92.35 %	90.21 %	98.54 %	97.22 %	93.70 %	90.48 %
F$_1$-measure	74.05 %	81.66 %	85.84 %	93.00 %	82.53 %	86.62 %

8 Summary

We analysed accuracy of a Named Entity Recognition (NER) algorithm based on Hidden Markov Model. The algorithm is limited to recognition and classification of Named Entities (NEs) representing persons. The algorithm was tested on two domain corpora and was compared with two simple base lines: a simple heuristic based on first upper case letter (Sec. 4.1) and a gazetteer-based algorithm (Sec. 4.2). The HMM was trained on the domain corpus of stock exchange reports and expressed promising results as tested on a separate subcorpus excluded from the training corpus: recall of 92% was high, but the precision of 63% was quite modest. However, precision could be easily increased by using a simple post-processing rules and was lifted up to 89% for the little cost of the 1% drop in recall. The results achieved on the domain corpus are promising, especially concerning very modest efforts required to prepare the training corpus.

We also tested the method portability to the second domain represented by the corpus of police reports. The HMM was trained on the corpus of stock exchange reports and next applied to the police reports. The obtained results were significantly worse: 28% precision and 48% recall, and after the rule-based filtering precision increased only to 64%. As we could expect, the machine learning method based only on probabilistic information like HMM, when applied to a limited training set fits more closely to the specific properties of the set.

In both cases the usage of the simple hand-written rules improved precision significantly. Concerning possible future NER algorithm, three types of knowledge should be combined in their construction: domain independent knowledge extracted by the means of machine learning from a general training corpus, domain specific knowledge extracted from a domain specific corpus by the means of machine learning and hand-written rules focused mainly on post-processing.

We plan to investigate applicability of different types of machine learning methods to different types of knowledge, e.g. it seems that methods focused on probability, like HMMs, are better suited for building domain specific components, while for capturing the domain independent aspects we need a method allowing for easier generalisation. We also plan to expand the size of the training corpus and the set of annotations of different NE types.

References

1. Piskorski, J.: Extraction of Polish named entities. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, pp. 313–316. ACL, Prague (2004)
2. Urbańska, D., Mykowiecka, A.: Multi-words Named Entity Recognition in Polish Texts. In: SLOVKO 2005 – Third International Seminar Computer Treatment of Slavic and East European Languages, Bratislava, Slovakia, pp. 208–215 (2005)
3. Mykowiecka, A., Kupść, A., Marciniak, M., Piskorski, J.: Resources for Information Extraction from Polish texts. In: Proceedings of the 3rd Language & Technology Conference (LTC 2007), Poznań, Poland (2007)
4. Graliński, F., Jassem, K., Marcińczuk, M., Wawrzyniak, P.: Named Entity Recognition in Machine Anonymization. In: Kłopotek, M.A., Przepiorkowski, A., Wierzchoń, A.T., Trojanowski, K. (eds.) Recent Advances in Intelligent Information Systems, pp. 247–260. Academic Publishing House Exit, San Diego (2009)
5. Graliński, F., Jassem, K., Marcińczuk, M.: An Environment for Named Entity Recognition and Translation. In: Màrquez, L., Somers, H. (eds.) Proceedings of the 13th Annual Conference of the European Association for Machine Translation, Barcelona, Spain, pp. 88–95 (2009)
6. Marcińczuk, M., Piasecki, M.: Pattern Extraction for Event Recognition in the Reports of Polish Stockholders. In: Proceedings of the Inter. Multiconference on Computer Science and Information Technology, Wisła, Poland, pp. 275–284 (2007)
7. Marcińczuk, M.: Pattern Acquisition Methods for Information Extraction Systems. Master thesis at Blekinge Tekniska Högskola, Sweden (2007)
8. Bikel, D. M., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a High-Performance Learning Name-finder. In: Proceedings of Conference on Applied Natural Language Processing (1997)
9. Zhou, G., Su, J.: Named Entity Recognition using an HMM-based Chunk Tagger. In: ACL 2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 473–480 (2002)
10. Carpenter, B.: Character Language Models for Chinese Word Segmentation and Named Entity Recognition. In: Proceedings of the 5th ACL Chinese Special Interest Group (SIGHan), Sydney, Australia (2006)
11. Malouf, R.: Markov models for language-independent named entity recognition. In: Proceedings of the Sixth Conference on Natural Language Learning, pp. 183–186 (2002)
12. Alias-i, LingPipe 3.9.0 (October 1 2008), http://alias-i.com/lingpipe
13. Marcińczuk, M., Piasecki, M.: Named Entity Recognition in the Domain of Polish Stock Exchange Reports. In: Kłopotek, M.A., Przepiórkowski, A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Systems, Siedlce, pp. 127–140 (2010)

Semantic Role Patterns and Verb Classes in Verb Valency Lexicon

Zuzana Nevěřilová

Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic
xpopelk@fi.muni.cz

Abstract. For Czech language there is large valency frame lexicon: VerbaLex. It contains verbs, slots related to the verbs and information about semantic roles each slot plays. This paper discusses observations made on VerbaLex frames related to verb classification. It shows that for particular classes of verbs (e.g. verbs describing weather) some semantic role patterns are typical. It also tries to reveal these patterns in not so obvious cases.

Currently, verb frames in VerbaLex are not interconnected. This paper outlines the way we can do such connections. We expect that verb frames of the same class or with the same semantic role patterns are semantically close and therefore propose similar types of interconnection. We expect to create relatively small set of inference rules that influence a large number of verb frames.

Keywords: Verbalex, valency lexicon, verb frames, FrameNet.

1 Introduction

Valency frame lexicon consists of following units:

- verb – a word and its synonyms describing an action, event or state
- verb frame – syntactic and semantic description of sentence constituents dependent on the verb
- slot – description of each dependent constituent

Valency frame lexicon serves not only as a syntactic description of verb dependent constituents, but also helps to describe or predict their semantic roles. We consider that the meaning of a sentence is composed by meanings of its constituents and *syntactic structure* the constituents form. Due to this we can use verb frames for semantic disambiguation. Moreover if we are able to semantically disambiguate sentences in a discourse, we will be able to put relations of known types between constituents in the sentences (e.g. cause–effect).

In this paper we study valency frame lexicon of Czech verbs VerbaLex. We construct the *semantic role patterns* and compare it with verb classification.

We in short introduce VerbaLex in Section 2. Section 3 defines and describes semantic role patterns in detail. In Section 4 we describe verb classes and evaluate the VerbaLex data w.r.t. semantic role patterns. In Section 5 we describe the generalization as the result of the two approaches. We discuss the possibility of interconnecting VerbaLex frames. Section 6 gives a conclusion and proposes future work.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 150–156, 2010.

2 VerbaLex

VerbaLex [1] is a valency frame lexicon built for Czech language. Currently it contains 19,360 verb frames for more than 10,000 verbs [2]. Semantic information is available on two levels:

- *semantic role* (also known as thematic role or thematic relation) that a sentence constituent plays w.r.t. the action or state. The concept is based on [3] and currently widely used (with some changes). VerbaLex contains 33 semantic roles such as agent, patient, location or substance.
- *semantic restriction* on a hypernym (e.g. person). This second level is related to WordNet's hypernym [4] (e.g. `person:1`, where `person` is a literal and `1` is the sense number).

Moreover a list of grammatical features such as preposition and grammatical case are present for each slot.

VerbaLex was built by lexicographers, independently of corpora information. It differs from VALLEX [5] mainly in its size and structure. VALLEX is closely related to Prague Dependency Treebank, while VerbaLex was built independently from it. VALLEX has no relation to WordNet, it contains only thematic roles (called *functors* in VALLEX). Moreover the function of these functors is different. Verb frames in both lexicons are not compatible.

3 Semantic Role Patterns

3.1 First Level Semantic Role Patterns

This paper concerns not single verbs, but groups of verbs that are expected to be semantically close. We define *1st level semantic role pattern* for a particular verb frame as a tuple $P = (R_1, \dots R_n)$. One of the elements of P is always the verb (marked as `VERB` later in this paper), other R_i are semantic roles assigned to this verb frame.

We made observations on the 1st level semantic role patterns and their frequency in VerbaLex. Table 1 shows most frequent patterns with example verbs. We can see that in some cases the verbs are semantically close while in other cases there is no perceivable semantic closeness. This feature significantly depends on the type of semantic role: the more specific it is, the more relationship among verbs we can observe. For example patterns containing communication (`COM`) group semantically close verbs, while patterns containing abstract object (`OBJ`) embody different groups of verbs. N.B. that only first level (semantic role) of VerbaLex is considered and no grammatical features are considered.

3.2 Second Level Semantic Role Patterns

Similarly to 1st level semantic role patterns we define *2nd level semantic role patterns* for a particular verb frame as a tuple of pairs $P = ((R_1, W_1), \dots (R_n, W_n))$. Here again R_i is a semantic role (one of them is the verb) and W_i is a WordNet hypernym that restricts the sentence constituent. Note that W_i has not to be present for every semantic role. In this case it is marked as ε.

Table 1. 12 most frequent 1st level semantic role patterns with number of occurrences (of appropriate verb frames) in VerbaLex

semantic role pattern	# of frames	example verbs	translation
(AG,VERB,PAT)	1,049	bodat, ovládnout, štěkat	sting, dominate, bark
(AG,VERB,OBJ)	866	bouchat, klovat, kácet	knock, (bird) peck, lumber
(AG,VERB,ACT)	788	detekovat, kazit, litovat	detect, destroy, sorry
(AG,VERB,PAT,ACT)	444	blahopřát, tázat se	compliment, ask
(AG,VERB,ART)	403	obarvit, kompilovat, koupit	color, compile, buy
(AG,VERB,LOC)	394	uzavřít, chvátat	close, rush
(AG,VERB,COM)	388	analyzovat, psát, klábosit	analyse, write, chat
(AG,VERB,STATE)	339	adaptovat se, dosáhnout, objasnit	adapt, achieve, clarify
(AG,VERB,KNOW)	297	bádat, konvertovat	research, convert
(AG,VERB,SUBS)	295	bagrovat, pít, vařit	dig, drink, cook
(AG,VERB,ENT)	279	krást, podobat se	steal, resemble
(AG,VERB,OBJ,OBJ)	261	doplnit, rozeznat	supplement, distinguish

Table 2 shows the most frequent second level semantic role patterns.

Table 2. 12 most frequent 2nd level semantic role patterns with number of occurrences (of appropriate verb frames) in VerbaLex

semantic role pattern	# of frames
((AG, person:1), (VERB, ε), (PAT, person:1))	800
((AG, person:1), (VERB, ε), (OBJ, object:1))	553
((AG, person:1), (VERB, ε), (ACT, act:2))	543
((AG, person:1), (VERB, ε), (PAT, person:1), (ACT, act:2))	299
((AG, person:1), (VERB, ε), (DPHR, ε)	242
((AG, person:1), (VERB, ε), (STATE, state:4))	224
((AG, person:1), (VERB, ε), (LOC, location:1))	176
((AG, person:1), (VERB, ε), (EVEN, event:1))	171
((AG, person:1), (VERB, ε)	170
((AG, person:1), (VERB, ε), (OBJ, object:1), (OBJ, object:1))	147
((AG, person:1), (VERB, ε), (PAT, person:1), (DPHR, ε))	134
((AG, person:1), (VERB, ε), (INFO, info:1))	127
((AG, person:1), (VERB, ε), (ART, artifact:1))	120
((AG, person:1), (VERB, ε), (PAT, person:1), (OBJ, object:1))	116

4 Verb Classes

Verb classes, defined "in terms of shared meaning components and similar syntactic behavior of words" [6] are useful because of generalizations. Since there are thousands of verbs we prefer processing whole classes instead of single verbs.

So far there are 5,638 verbs classified that makes about 25% of all verbs in the lexicon. The classification is based on [7] (VerbNet's classification is based on Levin's classes of English verbs), but adapted for Czech language.

Table 3 shows the relation between semantic role patterns and verb classes. Although there are many patterns for each verb class (only those with frequency greater than 5 are shown), we can see some similarities. For example, often a pattern is subset of another pattern in the same verb class. Or, in other words, a particular semantic relation is contained in (almost) every pattern for a particular verb class. Note that classified verbs were considered, therefore the number of occurrences is related to about 25% of the data.

Table 3. 10 most frequent verb classes with their semantic role patterns

no. of occurrences in VerbaLex	semantic role pattern	verb class
22	(AG,VERB,OBJ,SUBS)	butter-9
15	(AG,VERB,ART)	
15	(AG,VERB,OBJ)	
10	(AG,VERB,PAT,SUBS)	
9	(AG,VERB,PAT,PART,SUBS)	
9	(AG,VERB,SUBS)	
20	(AG,VERB,PAT)	judgement-33.2
11	(AG,VERB,PAT,ACT)	
9	(AG,VERB,PAT)	
8	(AG,VERB,ACT)	
6	(AG,VERB,ATTR)	
15	(AG,VERB,OBJ)	remove-10.1
9	(AG,VERB,OBJ,LOC)	
8	(AG,VERB,OBJ,OBJ)	
7	(AG,VERB,OBJ,LOC,LOC)	
7	(AG,VERB,PAT)	
15	(AG,VERB,OBJ)	bodyinternalmotion-49
9	(AG,VERB,PART)	
15	(AG,VERB,ACT)	want-32
10	(AG,VERB,PAT)	
6	(AG,VERB,ABS)	
6	(AG,VERB,OBJ)	
13	(AG,VERB,PAT,PART)	spank-18.3
12	(AG,VERB,PAT)	
11	(AG,VERB,PAT,INS)	
13	(AG,VERB,OBJ)	fill-9
8	(AG,VERB,PAT,ART)	
7	(AG,VERB,OBJ,OBJ)	
13	(AG,VERB,KNOW)	discover-82
11	(AG,VERB,ACT)	
11	(AG,VERB,STATE)	
11	(ENT,VERB)	animalsounds-38
8	(AG,VERB)	
10	(AG,VERB,PAT)	see-30.1
10	(AG,VERB,OBJ)	
6	(AG,VERB,EVEN)	

5 Generalization

In previous sections we have described in detail semantic role patterns and verb classes. The main difference is the manner they were acquired. In case of semantic role patterns, we deduce the semantic closeness only from the semantic roles of the verb dependent constituents. Therefore some of the verb groups are semantically closer than other.

Verb classes were made by linguists to group semantically close verbs together. Although the "behavior" of the verb was observed, the main decision feature is the meaning. We expect that both semantic role patterns and verb classes can serve to VerbaLex extension.

In case of 2nd level semantic role patterns we observe that the semantic closeness is more significant than if using only 1st level. For example verbs with the pattern `((AG,person:1),(VERB,`ε`),(PAT,person:1))` describe human interaction with another human(s). It is therefore meaningful to ask the following questions:

- was the action positive or negative for the agent (`AG`)?
- was the action positive or negative for the patient (`PAT`)?
- has the patient (`PAT`) to be at the same time on the same place as the agent (`AG`)?
- has the patient (`PAT`) to be in physical contact with the agent (`AG`)?

For verbs with the pattern `PART(bodypart:1)+VERB+PAT(person:1)` describing the action or state of body parts there are following expectations:

- the body part (`PART`) is part of the patient (`PAT`)
- there is something unusual with the patient's (`PAT`) body part (`PART`)

The following question is meaningful:

- was the action positive or negative for the patient (`PAT`)?

Similarly we can construct sets of expectations and meaningful questions for verb classes.

5.1 Extending VerbaLex with Inference Rules

VerbaLex is a frame-based lexical resource. It contains slots describing typical situations (in this case sentence constituents dependent on the verb), with restriction on their values (in this case WordNet hypernyms). Contrary to other frame-based resources such as FrameNet [8], VerbaLex frames are not related together, there is no hierarchy among the frames.

According to [9] we would like to extend VerbaLex with the interconnection of frames. At first, three relation types come on force: precondition, effect and decomposition. With the described generalization we can create small sets of inference rules for interconnecting large sets of verbs. These rules can look as in Figure 1.

The rules outlined on Figure 1 aim to transform VerbaLex from separate verb frames collection to a semantic network with inference as did Chang, Narayanan and Petruck within FrameNet [10]. Obviously the example above is just intuitive, not formalized.

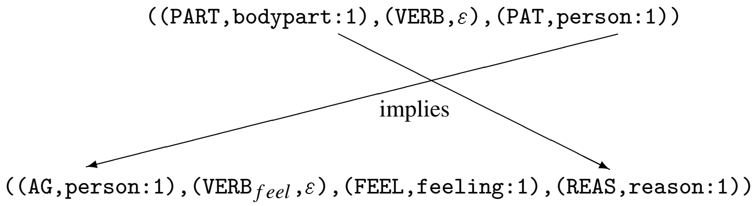

((PART,bodypart:1),(VERB,ε),(PAT,person:1))
implies
((AG,person:1),(VERB$_{feel}$,ε),(FEEL,feeling:1),(REAS,reason:1))

Fig. 1. A rule stating that someone is feeling something because of his/her bodypart action/state. Arrows show what sentence constituents refer to the same entity.

We also have to consider the loss and gain when processing groups of verbs instead of processing single verbs. The main advantage is the effectivity: with relatively small number of rules we can interconnect large number of frames. The disadvantage of this approach is that it cannot handle exceptions to these rules and it cannot handle cases when the verb is badly annotated (either it has bad roles in slots, or bad verb class). On the other hand, side effect of this work is in checking VerbaLex. We can see “suspicious” verb frames (e.g. those verbs that are in a particular class but do not follow patterns of other verbs of the same class) and check manually if they are annotated appropriately.

6 Conclusion and Future Work

This paper studies the structure of verb frames found in Czech valency frame lexicon VerbaLex. Two approaches of grouping verbs are shown: semantic role patterns extracted from VerbaLex and verb classification. We expect that verbs with same semantic role patterns are semantically close. Therefore, the patterns are compared to verb classes (based on VerbNet’s classification).

The purpose of this work is to make useful generalizations on verb groups. Due to these generalizations we can apply small number of inference rules on large number of verbs. Side effect of this work is verification of VerbaLex.

In the future we have to concentrate on formal representation of inference rules. Future work also concerns construction of rules that can interconnect the verb frames in VerbaLex. If this work succeeds, VerbaLex will acquire a new dimension of meaning representation for Czech language.

Acknowledgments

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536.

References

1. Hlaváčková, D.: Databáze slovesných valenčních rámců VerbaLex. Ph.D. thesis, Masarykova univerzita, Filozofická fakulta, Ústav českého jazyka (2007)
2. Hlaváčková, D.: Počet lemmat v synsetech VerbaLexu. In: After Half a Century of Slavonic Natural Language Processing, Brno, Czech Republic. Tribun EU (2009)

3. Gruber, J. S.: Studies in Lexical Relations. Ph.D. thesis, MIT, Cambridge, MA (1965)
4. Fellbaum, C.: WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press, Cambridge (1998)
5. Lopatková, M., Žabokrtský, Z., Benešová, V.: Valency Lexicon of Czech Verbs VALLEX 2.0. Technical Report 34, UFAL MFF UK (2006)
6. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: A Large-Scale Classification of English Verbs. Language Resources and Evaluation Journal 42, 21–40 (2008)
7. Schuler, K. K.: VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. Ph.D. thesis, Faculties of the University of Pennsylvania (2005)
8. Baker, C.F., Fillmore, C.J.: FrameNet 2009 (Online accessed July 30, 2009)
9. Nevěřilová, Z.: Exploring and Extending Czech WordNet and VerbaLex. In: Proceedings of the RASLAN Workshop 2009, Brno, Masaryk University, pp. 69–79 (2009)
10. Chang, N., Narayanan, S., Petruck, M.R.L.: From Frames to Inference. In: Proceedings of the First International Workshop on Scalable Natural Language Understanding (2002)

Opinion Mining by Transformation-Based Domain Adaptation

Róbert Ormándi[1], István Hegedűs[1], and Richárd Farkas[2]

[1] University of Szeged, Hungary
{ormandi,ihegedus}@inf.u-szeged.hu
[2] Research Group on Artificial Intelligence of the Hungarian Academy of Sciences, Hungary
rfarkas@inf.u-szeged.hu

Abstract. Here we propose a novel approach for the task of domain adaptation for Natural Language Processing. Our approach captures relations between the source and target domains by applying a model transformation mechanism which can be learnt by using labeled data of limited size taken from the target domain. Experimental results on several Opinion Mining datasets show that our approach significantly outperforms baselines and published systems when the amount of labeled data is extremely small.

1 Introduction

The generalization properties of most statistical machine learning approaches are based on the assumption that the samples of the training dataset come from the same underlying probability distribution than those that are used in the prediction phase of the model. Unfortunately – mainly in real-world applications – this assumption often fails. There are numerous Natural Language Processing tasks where plentiful labeled training databases are available from a certain domain, but we have to solve the same task using data taken from a different domain where we have only a small dataset. Manually labeling the data in the new domain is costly and inefficient. However, if an accurate statistical model from the source domain is present we can adapt it to the target domain [1].

Opinion Mining aims to automatically extract emotional cues from texts [2]. For instance it can classify product reviews according to the customers positive or negative polarity. Opinion Mining is a typical problem where the requirement for domain adaptation is straightforward as there exits numerous slightly different domains (e.g. different products are different domains) and the construction of manually labeled training data for each of them would be costly.

Here, we will define a general framework to directly capture the relations between domains. In order to experimentally evaluate our approach, Support Vector Machine (SVM) [3] was plugged into the framework and the approach was compared to a number of baseline algorithms and published results on Opinion Mining datasets.

2 Related Work

Numerous preliminary algorithms have been developed in the field of domain adaptation which roughly can be categorised into two mainstreams.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 157–164, 2010.

One of these types of methods tries to *model the differences between the distributions* of the source and target domains empirically. In the work proposed by Chelba et al. [4] the parameters of the maximum entropy model learned from the source domain as the means of a Gaussian prior was used during training a new model on target data. A different technique proposed by Daumé et al. [1] defines a general domain distribution that is shared between source and target domains. In this way, each source (target) example can be considered a mixture of source (target) and general distributions. Using these assumptions, their method was based on maximum entropy model and used the EM algorithm for training. Another approach was proposed by Daumé [5] where a heuristic nonlinear mapping function is used to map the data into a high dimensional feature space where a standard supervised learner can be employed in the area of domain adaptation.

The newer generation of domain adaptation algorithms are based on *defining new features for capturing the correspondence* between source and target domains [6,7]. In this way, the two domains appear to have very similar distributions, which enable effective domain adaptation. A more specific subtype of the above described algorithm family learns a *joint feature representation* for the source and the target domain where the corresponding marginal distributions are close to each other [8].

Theoretical results on domain adaptation have been also proposed [9]. For instance the work of Mansour et al. [9], the problem of multiple source domain adaptation was considered and was given theoretical results of the expected loss of combined hypotheses on the target domain.

3 Transformation-Based Domain Adaptation Approach

In this section we shall give a more precise formalism of the domain adaptation task and we will describe our approach in detail.

3.1 Domain Adaptation Task

In the current context of domain adaptation, we will assume that there are two feature spaces given – $\mathcal{D}_S$ and $\mathcal{D}_T$ – the "source domain" and the "target domain" feature spaces, respectively. We have two sets of labeled training samples, $S \subseteq \mathcal{D}_S$ and $T \subseteq \mathcal{D}_T$ as well ($|T| \ll |S|$). In addition we will assume that both the source domain and the target domain use the same label set. The labels[1] in both domains come from the $C = \{C_1, \ldots, C_l\}$ set and the $t : \mathcal{D}_S \cup \mathcal{D}_T \rightarrow C$ function assigns the *correct* class label to each sample from $\mathcal{D}_S$ and $\mathcal{D}_T$. The learning problem of the domain adaptation task is to find a $p_{\mathcal{D}_T} : \mathcal{D}_T \rightarrow C$ prediction function that achieves a high accuracy on the target domain.

3.2 Transformation-Based Approach

One of the main assumptions of the domain adaptation task is that there exists some kind of relation between the source domain and the target domain. Our idea is to try

[1] Our approach will focus on classification problems, but it can easily be extended to regression problems as well.

to model this relation, i.e. try to find a $\phi : \mathcal{D}_T \to \mathcal{D}_S$ transformation or target-source domain transformation. This transformation maps the samples from $\mathcal{D}_T$ into the feature space of $\mathcal{D}_S$.

More precisely, we look for a $\phi : \mathcal{D}_T \to \mathcal{D}_S$ transformation which minimizes the prediction error of each transformed sample taken from the training database of the target domain. Our idea is to utilize the $p_{\mathcal{D}_S} : \mathcal{D}_S \to C$ model (a prediction function on the source domain with a high prediction accuracy) directly for this task. Hence the following optimization problem was formed: $\min_\phi \quad E_{T,p_{\mathcal{D}_S}}(\phi) + Q \sum_{x \in T} \|\phi(x)\|$. Here $E_{T,p_{\mathcal{D}_S}}(\phi)$ is an error function which just depends on ϕ. If we can solve this optimization problem, we will get the prediction function of the target domain in the form $p_{\mathcal{D}_T}(x_0) = p_{\mathcal{D}_S}(\phi^*(x_0))$. Here the $\phi^* : \mathcal{D}_T \to \mathcal{D}_S$ mapping is the transformation which is the solution for the above-defined minimization task and $x_0 \in \mathcal{D}_T$ is an arbitrary sample from the target domain.

In this paper, we shall apply the following constraints on target-source domain mapping and on the two domains: $\mathcal{D}_S := \mathbb{R}^n$, $\mathcal{D}_T := \mathbb{R}^m$ and $\phi : \mathbb{R}^m \to \mathbb{R}^n, \mathbb{R}^m \ni x \mapsto Wx \in \mathbb{R}^n$, where $W \in \mathbb{R}^{n \times m}$. With these constraints, we will get the following specialized optimization task: $\min_{W,\, \|W\|=1} \quad E_{T,p_{\mathcal{D}_S}}(W)$. Here the regularization term is not necessary since it is replaced by the $\|W\| = 1$ constraint on the transformation matrix. This modification can be interpreted as the regularization term in the original form without weighting. To solve this optimization problem we can simply use a gradient descent-based optimization algorithm [10].

3.3 Support Vector Machine as Source Model

Here, the widely used SVM [3] classification method will be intorduced as the base prediction method[2] and the following error function:

$$E_{T,p_{\mathcal{D}_S}}(W) = \frac{1}{2} \sum_{x \in T} \left(t(x) - p_{\mathcal{D}_S}(Wx)\right)^2. \tag{1}$$

We assume that both the source domain and the target domain are labeled with the following labels: $C = \{-1, +1\}$ (binary classification). In this case the prediction function of the SVM classifier in our formalism is:

$$p_{\mathcal{D}_S}(Wx) = \sum_{s_k \in SV_S} \alpha_k t(s_k) K(s_x, Wx) + b. \tag{2}$$

Here SV_S denotes the set of support vectors that is the subset of the training database of the source domain, i.e. $\|SV_S\| \leq \|S\|$, s_k denotes the kth support vector, α_k is the learnt coefficient corresponding to s_k, the $b \in \mathbb{R}$ value is a learnt parameter of SVM as well and $K : \mathcal{D}_S \times \mathcal{D}_S \to \mathbb{R}$ is the kernel function over the source domain. The argument of the prediction function is Wx, which is the product of the transformation matrix W

[2] We derived and implemented Logistic Regression as source model with Cross Entropy error function as well. The description of this learner and the results achieved by it are available at http://www.inf.u-szeged.hu/rgai/~ormandi/DA2010_TSD_sup.pdf

and an arbitrary sample from the target domain $x \in \mathcal{D}_T$. Here the multiplying with W means the target-source domain mapping.

We decided to apply two commonly used kernel functions to compute the necessary gradient: the Polynomial kernel and the RBF kernel [3]. In Eq. (3) we can see the gradient of the error function applying the polynomial kernel. The form of the kernel is shown in this equation as well. The degree of the polynomial is denoted by d.

$$\begin{aligned} K_d(s_k, Wx) &= (s_k Wx)^d , \\ \nabla E_{T, p_{\mathcal{D}_S}, K_d}(W) &= -d \sum_{s_k \in SV_S} \alpha_k t(s_k) \cdot \\ \sum_{x \in T} \big(t(x) - p_{\mathcal{D}_S}(Wx)\big) & K_{d-1}(s_k, Wx)\, s_k x^T \end{aligned} \tag{3}$$

Similarly, in Eq. (4) we show the RBF kernel and the gradient of the error function using the RBF kernel. Here γ is a parameter of the RBF kernel.

$$\begin{aligned} K_\gamma(s_k, Wx) &= \exp\left(-\gamma \left\|s_k - Wx\right\|^2\right), \\ \nabla E_{T, p_{\mathcal{D}_S}, K_\gamma}(W) &= -2\gamma \sum_{s_k \in SV_S} \alpha_k t(s_k) \cdot \\ \sum_{x \in T} \big(t(x) - p_{\mathcal{D}_S}(Wx)\big) & K_\gamma(s_k, Wx)\,(s_k - Wx)\, x^T \end{aligned} \tag{4}$$

These gradients can be employed in the gradient descent-based algorithm. The whole learning systems will be denoted by PolyDML (using the Polynomial Kernel) and RBFDML (using the RBF Kernel and its gradient).

4 Experimental Results

In this section, the experimental results achieved on a synthetic dataset and real-world Opinion Mining tasks will be presented.

4.1 Evaluation Methodology

We hypothetised that domain adaptation is especially required when target training dataset is small, thus experiments using target training data with various sizes were carried out. In the case of extremely small datasets one evaluation per target domain size could not be trusted, thus for each size of the target domain we performed 10 runs and computed the average value of the elementary accuracy scores along with their variances.

Supervised SVMs trained on the target training data were employed as a baseline method (the usual choice in domain adaptation settings [5]). and SVMLight [3] was used as an SVM implementation.

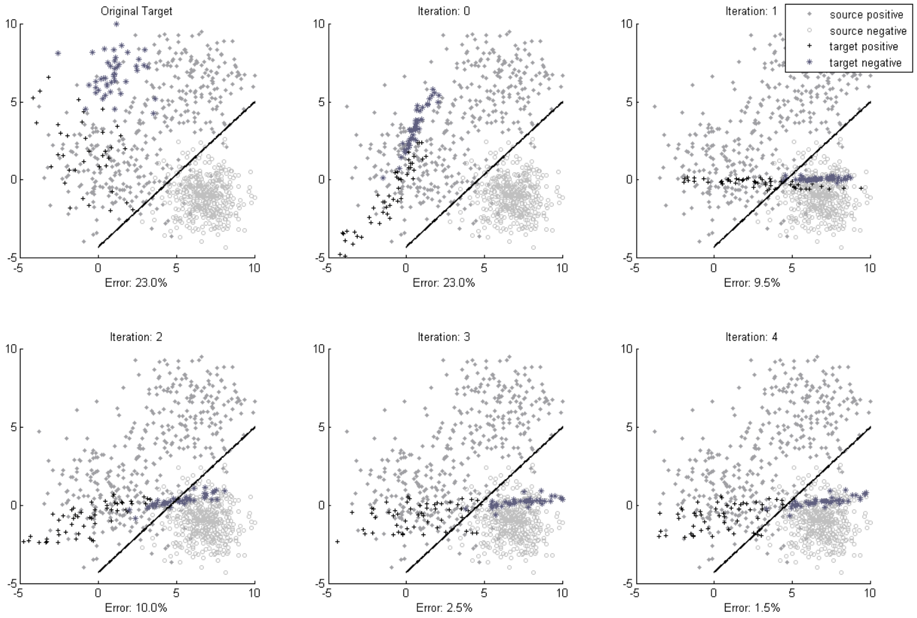


Fig. 1. The first 4 iterations of DML algorithm on Synthetic Database

4.2 Synthetic Database

To gain insight into the behaviour of our Transformation-based approach, we considered synthetically generated source and target domains. In order to visualize it, both domains were two dimensional. The positive samples of the source domain were generated based on the sum of two Gaussian distributions and the negative ones similarly, but using just one Gaussian distribution. We generated 1,000 samples and used only the first 800 of them as the training database of the source domain. The training and evaluation sets of the target domain were generated from the previously generated 1,000 samples by rotating them by 90 degrees and the same train-test split was employed.

In Fig. 1 we can see a sample run of the PolyDML algorithm on the synthetic database. We applied the Polynomial kernel with $d = 1$ (i.e. the Linear kernel) and set the C value of SVM to 1. The figure shows six different states of the algorithms. In each state we can see the data samples of the source domain and the classification boundary, which are constants. The first state shows the position of the original training samples of the target domain based on the samples taken from the source domain. The second state called "Iteration 0" shows the position of samples of the target domain which were transformed by applying a $W^{(0)}$ random transformation from the gradient descent-based algorithm proposed in section 3.2. The next four states show the first four iterations of the DML algorithm. For each state we also included the error measured on the target train dataset. As one can see, in the initial states (i.e. in the first two states) the error rate is quite high, but in the first four iterations the error rate decreases fast

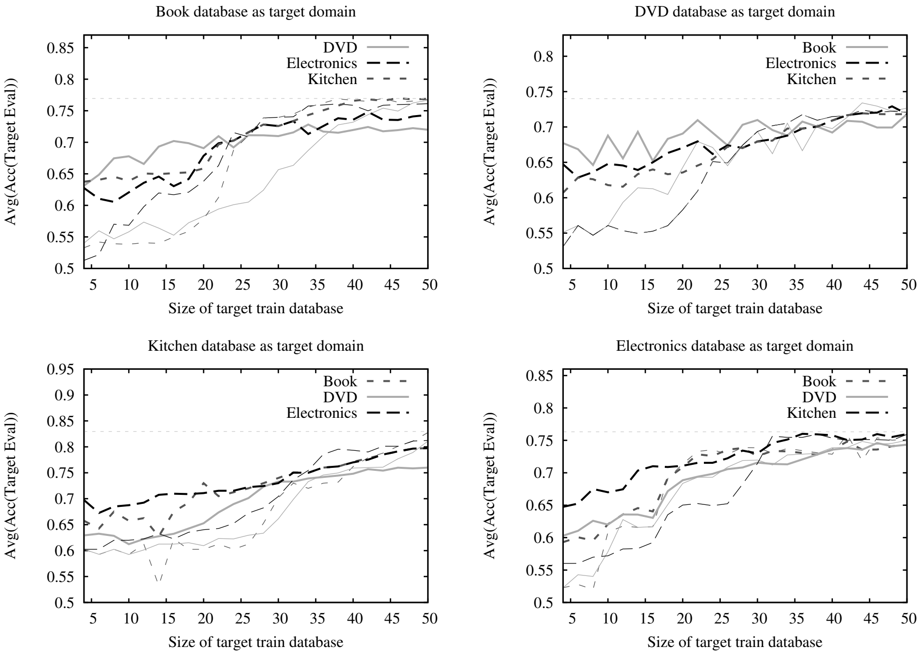


Fig. 2. The average accuracies of RBFDML algorithm using different sizes of subsets of the target domains of Multi-Domain Sentiment Dataset. (In each subfigure each of thinner lines denotes the corresponding baseline result, and the result denoting by a horizontal line accords to the full sized target train dataset.)

and almost monotonically. PolyDML significantly outperforms the supervised baseline as well ($Error = 17.0\%$).

4.3 Results on Multi-domain Sentiment Dataset

Our Transformation-based method was evaluated on Opinion Mining datasets [11] as well. These datasets contains product reviews taken from Amazon.com for four product types (domains), namely: books, DVDs, electronics and kitchen appliances. Originally the attributes of instances of each dataset were the term frequencies of the words of the corresponding review texts, and the labels of the instances were generated from the rating of the reviews. More precisely, reviews with rating ≥ 3 were considered as positive, while those with rating < 3 were labeled negative (binary classification problem). The datasets of each domain were balanced, all of them having 1,000 positive and 1,000 negative samples with a very high dimension (about 5,000 dimensions), because each different word in a review generates a dimension in the database.

We split the datasets of each domain into two parts in a random way (80% training set and 20% evaluation set). Then we performed a feature selection step, selecting the attributes where the InfoGain score was positive on the train set and performed a Principle Component Analysis (PCA) on each training dataset. The feature dimensionality reduction steps found on the training sets were then applied to the evaluation sets.

Since we had four different domains, we investigated all the possible 12 domain adaptation tasks. The results of this are summarized in Fig. 2. Each sub-figure shows the results of RBFDML and the corresponding supervised methods (baselines) and – with a horizontal line – the result of the direct method applying the SVM source model which uses the *full* training dataset of the target domain. This is independent of the values of the x axis and can be viewed as the "limit values" of the corresponding results of direct methods. At each point in the sub-figures we can see average accuracy scores of 10.

As can be seen in Fig. 2, when we use limited-sized datasets from the target domain, the proposed methods can achieve a significantly higher accuracy than the baseline methods. The reason for this phenomenon might be that the baseline could not made valid generalization from the small number of samples – since the database of the target domain might not contain enough information to build a well-generalizing model – but the transformation-based approach uses the well-generalized source model which helps the generalization of the final transformation-based model.

Structural Correspondence Learning (SCL) is a domain adaptation approach [11] which has published results on the Opinion Mining datasets we used. In comparison with its results, our approach achived better accuracy scores 10 times compared to the base SCL, and 7 times compared its extended version (SCL-MI).

5 Conclusions

In this paper, we presented our novel, transformation-based approach for handling the task of domain adaption. We have described two instances of our main algorithm and experimentally showed that – applying them to a real world dataset in 12 different scenarios – our methods outperform the baseline approaches (direct methods) and published results of the same dataset.

Our experimental results proved that it is possible to train models for the target domain that uses a very limited number of labeled samples taken from this domain. This is true as well in those cases when there are enough samples, but baseline methods cannot generalize well using such samples. On the other hand, our approach has a key advantage against other domain adaptation procedures as it does not require access to the source data just to a trained source model which can be crucial in several cases (e.g. privacy issues).

In the near future we would like to investigate our general approach with other learning models.

References

1. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. J. Artif. Int. Res. 26, 101–126 (2006)
2. Kobayashi, N., Inui, K., Matsumoto, Y.: Extracting Aspect-Evaluation and Aspect-of Relations in Opinion Mining. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1065–1074 (2007)

3. Joachims, T.: Making Large-Scale Support Vector Machine Learning Practical, pp. 169–184 (1999)
4. Chelba, C., Acero, A.: Adaptation of Maximum Entropy Capitalizer: Little Data Can Help a lot. Computer Speech & Language 20, 382–399 (2006)
5. Daumé III, H.: Frustratingly Easy Domain Adaptation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 256–263 (2007)
6. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of Representations for Domain Adaptation. In: Advances in Neural Information Processing Systems, vol. 20, MIT Press, Cambridge (2007)
7. Gupta, R., Sarawagi, S.: Domain Adaptation of Information Extraction Models. SIGMOD Rec. 37, 35–40 (2008)
8. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain Adaptation via Transfer Component Analysis. In: IJCAI, pp. 1187–1192 (2009)
9. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain Adaptation with Multiple Sources. In: NIPS, pp. 1041–1048 (2008)
10. Snyman, J.A.: Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms (Applied Optimization). Springer, New York (2005)
11. Blitzer, J., Dredze, M., Pereira, F.: Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 440–447 (2007)

Improving Automatic Image Captioning Using Text Summarization Techniques

Laura Plaza[1], Elena Lloret[2], and Ahmet Aker[3]

[1] Universidad Complutense de Madrid
C/Prof. José García Santesmases, s/n 28040 Madrid, Spain
lplazam@fdi.ucm.es
[2] University of Alicante, Apdo. de correos, 99, E-03080 Alicante, Spain
elloret@dlsi.ua.es
[3] University of Sheffield, Regent Court, 211 Portobello St., Sheffield, S1 4DP, UK
a.aker@dcs.shef.ac.uk

Abstract. This paper presents two different approaches to automatic captioning of geo-tagged images by summarizing multiple web-documents that contain information related to an image's location: a graph-based and a statistical-based approach. The graph-based method uses text cohesion techniques to identify information relevant to a location. The statistical-based technique relies on different word or noun phrases frequency counting for identifying pieces of information relevant to a location. Our results show that summaries generated using these two approaches lead indeed to higher ROUGE scores than n-gram language models reported in previous work.

Keywords: automatic image captioning, multi-document summarization, graph-based and knowledge-based text summarization.

1 Introduction

The number of images with location information is growing exponentially with the rapid development of online photo sharing services and increasing prevalence of camera phones with embedded GPS and compass. Additionally, many legacy photographs and other images are stored or tagged by place names or contain minimal captions that include geographical information. In all these cases the small or non-existent amount of textual information associated with the image is of limited usefulness for image indexing, organization, and/or search. What would be useful is a means to automatically generate or augment captions for images from their geo-referencing information.

Aside from application to image indexing, organization and search, the capability to automatically caption geo-referenced images has further potential applications. It could, for instance, help users gain quick access to the information they seek about a place of interest just by taking its picture. Such textual information could also be used by a journalist who is planning to write an article about a building, or by a tourist who seeks further places to visit nearby.

Attempts towards automatic generation of image captions have been previously reported. Deschacht & Moens [1] and Mori et al. [2] generate image captions

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 165–172, 2010.

automatically by analyzing image-related text from the immediate context of the image, e.g. the surrounding text in HTML documents. The authors identify named entities and other noun phrases in the image-related text and assign these to the image as captions. Other attempts towards automatic generation of image captions generate captions based on the immediate textual context of the image with or without consideration of image related features such as colour, shape or texture [1,2,3,4,5,6,7,8,9]. However, Marsch & White [10] argue that the content of an image and its immediate text have little semantic agreement and this can, according to Purves et al. [11], be misleading to image retrieval. Furthermore, these approaches assume that the image has been obtained from a document. In cases where there is no document associated with the image, which is the scenario we are principally concerned with, these techniques are not applicable.

Following the general approach proposed by Aker and Gaizauskas [12], in this paper we describe different methods for automatic image captioning starting with only a set of place names pertaining to an image – for example $\langle\{$St. Paul's, London$\}\rangle$. Place names can be obtained automatically given GPS coordinates and compass information using techniques such as those described in Xin et al. [13] – that task is not the focus of this paper.

Aker and Gaizauskas [12] have argued that humans appear to have a conceptual model of what is salient regarding a certain object type (e.g. church, bridge, etc.) and that this model informs their choice of what to say when describing an instance of this type. They also experimented with representing such conceptual models using n-gram language models derived from corpora consisting of collections of descriptions of instances of specific object types (e.g. a corpus of descriptions of churches, a corpus of bridge descriptions, and so on) and reported results showing that incorporating such n-gram language models as a feature in a feature-based extractive summarizer improves the quality of automatically generated summaries. However, the authors report that the quality of language model biased summaries was still not at satisfactory level.

In this paper we experiment with two different approaches to generate summaries related to images: a graph-based and a statistical-based approach. The graph-based method enriches the words in the documents with further concepts and relations from WordNet to better capture syntactic different but semantically similar feature descriptions used to described places. The statistical-based approach concentrates on long descriptive noun phrases frequency count to identify salient feature descriptions. Our results show that our methods indeed score better than the ones reported in Aker and Gaizauskas [12].

In the following we first describe the set of images, their model summaries and the retrieval of related web-documents (section 2). In Section 3 we present the summarizers used to caption images. We discuss the results of evaluating automatic summaries against the human created captions in Section 4, and draw conclusions and future lines of work in Section 5.

2 Corpus

For evaluation we use the image collection described in Aker and Gaizauskas [14]. The image collection contains 308 different images with manually assigned place names. For each image there are up to four short descriptions or model summaries. The model summaries were created manually based on image descriptions taken from *VirtualTourist* and contain a minimum of 190 and a maximum of 210 words.

To generate automatic captions for the images, Aker and Gaizauskas [12] automatically retrieved the top ten related web-documents for each image using the Yahoo! search engine and the place name associated with the image as a query. The text from these documents was extracted using an HTML parser and passed to their summarizer. We also used these documents to generate image captions.

3 Summarizers

3.1 A Semantic-Graph Based Summarizer

The summarizer has been already presented in previous work and evaluated in two different domains: news items and biomedical papers [15]. In this paper, we focus on image caption generation. First, it should be noted that the system was not originally designed to deal with multi-document summarization. To overcome this shortcoming, we simply merge all documents about the same topic into a single document, and run the summarizer over it. After producing the summary, we apply a textual entailment module to detect and remove redundancy [16]. The summarizer first applies a shallow preprocessing over the document, including sentence detection, POS tagging and removing stopwords and high frequency terms. It next translates the text in the document to WordNet concepts, using the *lesk* algorithm (as implemented in the WordNet Sense Relate package [17]) to disambiguate the meaning of each term in the document according to its context. After that, the resulting WordNet concepts are extended with their hypernyms, building a graph representation for each sentence in the document, where the vertices represent distinct concepts in the sentence and the edges represent *is-a* relations. The system then merges all the sentence graphs into a single document graph, which is extended with a further semantic relation, so that every pair of leaf vertices whose similarity (calculated in terms of WordNet concepts gloss overlaps, using the *WordNet Similarity* package [18]) exceeds a certain threshold. Each edge in the document graph is assigned a weight that is directly proportional to the depth in the hierarchy of the nodes that it links (that is, the more specific are the concepts connected by a link, the more weight is assigned to it).

Once the graph is built, the vertices are ranked according to their *salience* or prestige. The salience of a vertex is calculated as the sum of the weight of the edges connected to it. The top n vertices are grouped into *Hub Vertices Sets (HVS)*, which represent sets of concepts strongly related in meaning. These will constitute the centroids of the clusters. A *degree-based clustering* method [19] is then executed over the graph and, as a result, a variable number of clusters or subgraphs are obtained. The working hypothesis is that each of these clusters represents a different *subtheme* or topic within the document, and that the most central concepts in a cluster (the so called HVS) give the necessary and sufficient information related to its topic. The process continues by calculating the similarity between all the sentence graphs and each cluster. To this aim, a non-democratic vote mechanism [20] is used, so that each vertex (v_k) of a sentence (S_j) gives to each cluster (C_i) a different number of votes (w_i, j) depending on whether v_k

belongs or not to the HVS of that cluster. The similarity is computed as the sum of the votes given by all vertices in the sentence to each cluster. Finally, under the hypothesis that the cluster with more concepts represents the main theme in the document, and hence the only one that should contribute to the summary, the *N* sentences with greater similarity to this cluster are selected.

3.2 A Statistical-Based Summarizer

In previous work [21], different techniques were shown to be appropriate for the text summarization task. More specifically, three features were analyzed (term-frequency, textual entailment, and the code quantity principle), and the performance of several approaches employing such features on their own, as well as in combination, was investigated. After the research, it was concluded that the combination of all the techniques within the same approach led to the best results, outperforming by approximately 10% the best system in DUC 2002. Therefore, in this paper, we follow the same idea and we take an improved version of this approach as the basis for generating text summaries. However, it should be noted that this approach has been only evaluated over newswire, whereas in this research we focus on a completely different type of documents, *image captions*, which can be considered as one of the many new textual genres born with the Web 2.0. Besides the general assessment of the whole statistical-based summarization approach, the use of such corpus will also allow us to analyze whether the suggested techniques could be domain-independent or not.

Next, each of the features are briefly described. First, a textual entailment tool [16] is used to detect redundant information in the document and, as a consequence, the repeated information is removed. The other features, term-frequency and the code quantity principle[1] are used to measure the importance of each sentence in the document, thus assigning a score to each one based on the frequency of the words that a noun phrase contains and normalizing this value with respect to the number of total noun phrases. A detailed description of these features can be found in [21]. It is worth stressing upon the fact that the summarization approach was originally working only for single-document summarization. In order to allow it to deal with several documents, we adapted it in the same way we did in the semantic-graph based approach previously explained.

The summarization process starts with an initial stage where some basic preprocessing is carried out, which includes sentence segmentation, tokenization, part-of-speech tagging, and *stopwords* removal. At this stage, the frequency of each remaining word is counted and stored. Then, the textual entailment module is run over the documents in order to detect potential sentences with repeated information. Further on, once we have the text without redundant information, a relevance sentence detection stage computes a score for each sentence, based on the frequency of the words that appear in a noun phrase. Therefore, sentences not only with longer noun phrases but also with the most frequent words within these noun phrases are considered more important, and consequently, are assigned a higher score. This score is normalized by the number of noun

[1] The code quantity principle is a linguistic theory that proves the existence of a proportional relation between the importance of a piece of information and the number of text units it contains [22]. In our work, noun phrases are the text units taken into account.

phrases a sentence has. Finally, the summarization selects the highest ranked sentences, and presents them in the same order as they appeared in the original documents, in a preliminary attempt of maintaining the coherence of the text. The final summary is made up from these selected sentences.

4 Results

4.1 Experimental Framework

To evaluate both approaches, we use the image caption collection described in Section 2. We generate 200-words long summaries for the images from this collection, each of one is described by ten different documents, and compare the automatic summaries against the model summaries written by humans.

Following the Document Understanding Conferences [23], the ROUGE evaluation metric [24] is used for assessing the summarizers. ROUGE compares automatically generated summaries (called *peers*) against human-created summaries (called *models*), and computes a set of different measures to estimate content coverage in an automatically generated summary. In particular, we compute ROUGE-2 and ROUGE-SU4 recall scores. In short, ROUGE-2 evaluates bi-gram co-occurrence between the peer and model summaries, while ROUGE-SU4 allows bi-grams to have intervening word gaps no larger than four words.

As baseline we generate summaries using the Wikipedia article describing each image, from which we select the first 200 words. We look at these summaries as a difficult goal to achieve: first, it must be taken into account that these articles have been created by humans; second, the first paragraph in a Wikipedia article is usually just a summary of the entire document content; and third, Wikipedia articles almost exclusively contain salient information to the subject matter, and so do not present other information somehow related to the topic but not important (e.g. nearby hotels, restaurants, transport services, or even advertising).

4.2 Results and Discussion

Table 1 shows the ROUGE-2 and ROUGE-SU4 recall values for the Wikipedia baseline summaries as well as for the two suggested approaches: *semantic-graphs* and *statistical-based*. It also includes the best results of the n-gram *language models*, as reported in [12].

Table 1. ROGUE results for the different summarization approaches

	Wikipedia	**Semantic-graphs**	**Statistical-based**	**Language models**
Rouge-2	0.096***	0.089*	0.086	0.071
Rouge-SU4	0.142***	0.142***	0.134	0.119

It can be observed that both systems (semantic-graphs and statistical-based) achieve better results than language models in both ROUGE metrics. Besides, as expected, Wikipedia summaries significantly outperform the other summarizers (Wilcoxon Signed Ranks test[2]). However, the difference between Wikipedia and our two summarizers is less than we would have anticipated, specially in the ROUGE-SU4 score, which seems to indicate that both approaches provide a good approximation to the problem of summarizing information related to tourist images. Regarding the comparison between the two approaches presented here, the semantic-graph based method obtains significantly better for ROUGE-2 and ROUGE-SU4 score, for different confidence intervals.

In the remaining of the section, we will try to elucidate the reasons for the unfavorable differences between the summaries generated by our systems and Wikipedia summaries. Regarding the graph-based approach, the main problem is directly related to the type of the documents to summarize: in most of these documents, the salient information is concerned with proper nouns describing monuments, cities, beaches, etc., that are not likely to be found in WordNet (e.g. *Sacre Coeur*, *Santorini* or *Ipanema*). If no concept is found in the ontology for these terms, the document graph will be inevitably losing essential information to identify the topics covered in the document.

As far as the statistical-based approach is concerned, the main problem lies also in the nature of the corpus. Most documents in the corpus contain sentences with a high number of noun phrases, but which are unrelated to the topic (e.g *"Mahogany, Maple, crown mouldings, multiple Viking ovens, Sub-Zero refrigerators, antique... "*). According to the code quantity principle feature, these types of sentences are scored higher, thus being considered relevant to incorporate them to the summary. In these cases, the quality of the generated summaries is directly affected by these sentences.

5 Conclusion

In this paper we presented two different approaches – a semantic graph-based and a statistical-based approach – for automatically generating image captions from several documents retrieved from the Internet. The former takes into consideration the salience of the WordNet concepts in the text to identify important contents. The latter relies on long descriptive noun phrases together with the frequency of terms to identify relevant information. The results of both systems are highly satisfactory. They compare positively with previous approaches and their ROUGE scores are not far from those of the Wikipedia summaries.

The type of documents in hand, most of them extracted from tourist information websites, makes automatic summarization even more challenging than in other domains. In most of these documents, only a few information is relevant to the image, while the rest can be considered as noisy information (e.g. nearby hotels and other tourist services, advertisements from the website that hosts the information...). Besides, these documents are highly redundant.

However, the results reported show that there is room for improvement. In the future, we plan to overcome the limitations of our approaches that have been identified after the analysis of the results obtained. Incorporating some module able to identify the noisy information in the documents and filter it out would undoubtedly beneficial for both

[2] We use the following conventions for indicating significance level in the tables: *** = $p < .0001$, ** = $p < .001$, * = $p < .05$ and no star indicates non-significance.

systems. In the case of the statistical-based approach, a query-focused summarization approach would be necessary to identify only sentences talking about the topic (i.e. the place).

Acknowledgments

This research is funded by the Spanish Government through the FPU program and the projects TIN2009-14659-C03-01, TSI 020312-2009-44, and TIN2009-13391-C04-01; by Conselleria d'Educació – Generalitat Valenciana (grant no. PROMETEO/2009/119 and ACOMP/2010/286); and the FPI program (BES-2007-16268) from the Spanish Ministry of Science and Innovation (project TEXT-MESS (TIN2006-15265-C06-01)). It is also funded by the TRIPOD project supported by the European Commission (Contract No. 045335).

References

1. Deschacht, K., Moens, M.: Text Analysis for Automatic Image Annotation. In: Proc. of the 45th ACL (2007)
2. Mori, Y., Takahashi, H., Oka, R.: Automatic word assignment to images based on image division and vector quantization. In: Proc. of RIAO 2000: Content-Based Multimedia Information Access (2000)
3. Barnard, K., Forsyth, D.: Learning the semantics of words and pictures. In: International Conference on Computer Vision, vol. 2, pp. 408–415 (2001)
4. Duygulu, P., Barnard, K., de Freitas, J., Forsyth, D.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Seventh European Conference on Computer Vision (ECCV), vol. 4, pp. 97–112 (2002)
5. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D., Jordan, M.: Matching words and pictures. The Journal of Machine Learning Research 3, 1107–1135 (2003)
6. Pan, J., Yang, H., Duygulu, P., Faloutsos, C.: Automatic image captioning. In: IEEE International Conference on Multimedia and Expo., vol. 3 (2004)
7. Feng, Y., Lapata, M.: Automatic Image Annotation Using Auxiliary Text Information. In: Proc. of ACL 2008, Columbus, Ohio (2008)
8. Satoh, S., Nakamura, Y., Kanade, T.: Name-It: Naming and Detecting Faces in News Videos. IEE Multimedia 6, 22–35 (1999)
9. Berg, T., Berg, A., Edwards, J., Forsyth, D.: Whos in the Picture? In: Proc. Of Advances in Neural Information Processing Systems Conference (2005)
10. Marsh, E., White, M.: A taxonomy of relationships between images and text. Journal of Documentation 59, 647–672 (2003)
11. Purves, R., Edwardes, A., Sanderson, M.: Describing the Where – Improving Image Annotation and Search Through Geography. In: 1st Intl. Workshop on Metadata Mining for Image Understanding (2008)
12. Aker, A., Gaizauskas, R.: Summary Generation for Toponym-Referenced Images using Object Type Language Models. In: International Conference on Recent Advances in Natural Language Processing, RANLP (2009)
13. Fan, X., Aker, A., Tomko, M., Smart, P., Sanderson, M., Gaizauskas, R.: Automatic Image Captioning From the Web For GPS Photographs. In: Proc. of the 11th ACM SIGMM International Conference on Multimedia Information Retrieval (2010)

14. Aker, A., Gaizauskas, R.: Model Summaries for Location-Related Images. In: Proc. of the 7th conference on International Language Resources and Evaluation (2010)
15. Plaza, L., Díaz, A., Gervás, P.: Concept-Graph Based Biomedical Automatic Summarization Using Ontologies. In: Coling 2008: Proc. of the 3rd Textgraphs Workshop on Graph-Based Algorithms for NLP, pp. 53–56 (2008)
16. Ferrández, O., Micol, D., Muñoz, R., Palomar, M.: A Perspective-Based Approach for Solving Textual Entailment Recognition. In: Proc. of the ACL PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 66–71 (2007)
17. Patwardhan, S., Banerjee, S., Pedersen, T.: Senserelate: targetword: a Generalized Framework for Word Sense Disambiguation. In: Proc. of the ACL 2005 on Interactive Poster and Demonstration Sessions, Morristown, NJ, pp. 73–76 (2005)
18. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet: similarity – Measuring the Relatedness of Concepts. In: Proc. of the Nineteenth National Conference on Artificial Intelligence (AAAI 2004), pp. 1024–1025 (2004)
19. Erkan, G., Radev, D.: Lexrank: Graph-Based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research 22, 457–479 (2004)
20. Yoo, I., Hu, X., Song, I.Y.: A Coherent Graph-Based Semantic Clustering and Summarization Approach for Biomedical Literature and a New Summarization Evaluation Method. BMC Bioinformatics 8 (2007)
21. Lloret, E., Palomar, M.: A Gradual Combination of Features for Building Automatic Summarisation Systems. In: Proc. of the 12th International Conference on Text, Speech and Dialogue, pp. 16–23 (2009)
22. Givón, T.: Functional-Typological Introduction, II. John Benjamins, Amsterdam (1990)
23. Dang, H.: Overview of DUC 2005. In: DUC 2005 Workshop at HLT/EMNLP (2005)
24. Lin, C.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proc. of the Workshop on Text Summarization Branches Out (WAS 2004), pp. 25–26 (2004)

Perplexity of n-Gram and Dependency Language Models*

Martin Popel and David Mareček

Charles University in Prague, Institute of Formal and Applied Linguistics
{popel,marecek}@ufal.mff.cuni.cz

Abstract. Language models (LMs) are essential components of many applications such as speech recognition or machine translation. LMs factorize the probability of a string of words into a product of $P(w_i|\mathbf{h}_i)$, where $\mathbf{h}_i$ is the context (history) of word w_i. Most LMs use previous words as the context. The paper presents two alternative approaches: *post-ngram LMs* (which use following words as context) and *dependency LMs* (which exploit dependency structure of a sentence and can use e.g. the governing word as context). Dependency LMs could be useful whenever a topology of a dependency tree is available, but its lexical labels are unknown, e.g. in tree-to-tree machine translation. In comparison with baseline interpolated trigram LM both of the approaches achieve significantly lower perplexity for all seven tested languages (Arabic, Catalan, Czech, English, Hungarian, Italian, Turkish).

1 Introduction

Language models (LMs) are essential components of many applications such as speech recognition or machine translation (MT). LM is a statistical model that assigns a probability to every sentence s which is represented as a sequence of m words, i.e. $P(s) = P(w_1, \ldots, w_m)$. LMs factorize this joint probability into a product of conditional probabilities in form $P(w_i|\mathbf{h}_i)$, where h_i is the context (traditionally called history) of word w_i. Most LMs (e.g. standard n-gram LMs, maximum entropy LMs [1], factored LMs [2] and even some grammar-based LMs [3]) use previous words as the context, so the joint probability can be computed in a left-to-right manner:

$$P(s) = \prod_{i=1 \ldots m} P(w_i|w_1, \ldots, w_{i-1}). \quad (1)$$

N-gram LMs consider only the last $n-1$ words using so-called $(n-1)^{\text{th}}$ order Markov property, i.e. $\mathbf{h}_i \equiv (w_{i-n}, \ldots, w_{i-1})$; [1]

$$P_{ngram}(s) = \prod_{i=1 \ldots m} P(w_i|\mathbf{h}_i) = \prod_{i=1 \ldots m} P(w_i|w_{i-n}, \ldots, w_{i-1}). \quad (2)$$

However, we can use other factorization orderings instead of left-to-right. For example, in right-to-left ordering we use so-called *post-ngrams* (post-bigram, post-trigram, etc.) as the context, i.e. $\mathbf{h}_i \equiv (w_{i+1}, \ldots, w_{i+n-1})$;

* The work on this project was supported by the grants GAUK 116310, GA201/09/H057, FP7-ICT-2009-4-247762, and FP7-ICT-2007-3-231720.

[1] For simplicity, artificial start-of-sentence tokens are usually inserted before each sentence, in other words, for $i < 1$ we define $w_i \equiv$ `<NONE>`.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 173–180, 2010.

$$P_{post\text{-}ngram}(s) = \prod_{i=1...m} P(w_i|\mathbf{h}_i) = \prod_{i=1...m} P(w_i|w_{i+1}, \ldots, w_{i+n-1}). \quad (3)$$

Generally, we can define a directed acyclic graph (DAG) on words of the given sentence, and define $\mathbf{h}_i \equiv (w_j : (j, i) \in Edges(DAG))$.[2]

The key topic of this paper is a comparison of LMs based on different DAGs.[3] We are particularly interested in DAGs which are based on dependency parsing; we call the resulting language models *dependency LMs*.

In Sect. 2, we briefly summarize related work on dependency LMs. Afterwards, we describe possible ways of designing dependency LMs (Sect. 3) and exploiting additional context attributes (Sect. 4). Experiments are reported in Sect. 5 and concluding discussion is given in Sect. 6.

2 Related Work on Dependency LMs

There are papers (e.g. [3,4]) that use the term *dependency language model* (with different meanings though), but we are not aware of any universal definition of the term. Nevertheless, the common idea behind the term is to use some kind of dependency trees (such as those used in CoNLL shared tasks [5]) for language modeling. Hereinafter, *parent* denotes the governing word (and the corresponding tree node), similarly *children* denotes the dependent words (modifiers).

There are many possible ways how to exploit dependency trees in dependency LMs. Chelba et al. [3] use them in conventional left-to-right factorization ordering – briefly, the context considered for a word comprises the preceding bigram and a *link stack*, which is a list of words that precede the current word, but their parent does not. Shen et al. [4] compute the probability of a tree (which represents the given sentence) using probabilistic distributions P_L and P_R for left and right side generative probabilities respectively and P_T for a probability of a word being the root. The probability of a word is conditioned by its parent and also siblings that lie between the word and the parent. See Fig. 1 and Formula 4 for illustration.

$$\begin{aligned} Prob = & P_L(the|boy\text{-as-parent}) \\ & \times P_L(boy|will, find\text{-as-parent}) \times P_L(will|find\text{-as-parent}) \\ & \times P_T(find) \\ & \times P_R(it|find\text{-as-parent}) \times P_R(interesting|it, find\text{-as-parent}) \end{aligned} \quad (4)$$

Most related to our experiments is a research by Charniak [6], who defines two LMs based on his immediate-head parser. The first LM is called *bihead* – the probability of a

[2] Performing the factorization in DAG's topological ordering ensures, we condition always only on words whose probabilities have been already computed. Note that n-gram LMs are a special case of this generalized DAG-based LM – $Edges(DAG_{ngram}) = \{(j, i) : j \in \{i - n + 1, ..., i - 1\}\}$. Likewise, the before-mentioned post-ngram LMs are a special case of DAG-based LM – $Edges(DAG_{post\text{-}ngram}) = \{(j, i) : j \in \{i + 1, ..., i + n - 1\}\}$.

[3] The quality of LMs is measured by *cross-entropy* $H = -(1/|T|) \sum_{i=1}^{|T|} log_2 P(w_i|\mathbf{h}_i)$, where T is test data. For convenience, we report *perplexity* ($PP = 2^H$), as it is usual in literature. Lower perplexity implies better LM.

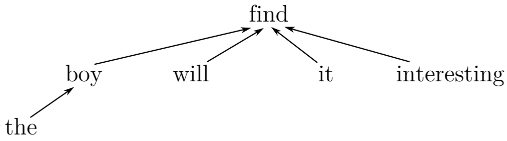


Fig. 1. Dependency tree of sentence “the boy will find it interesting”

word is conditioned by its parent, similarly to our model wp as it is defined in Sect. 3.1. The second LM is called *trihead* – it conditions by word’s parent and grandparent, similarly to our model wp,wq. Charniak reports 22% improvement in perplexity over trigram LM.[4]

3 Designing Dependency LMs

In Sect. 1 we introduced DAG-based LMs. DAGs of dependency LMs can be constructed from the dependency tree of a given sentence s using several methods. Actually, we need just the topology T of the dependency tree.[5]

3.1 Model wp (Word Form of Parent)

The simplest method for constructing dependency LMs is to use T itself with edges directed from parent to child as the DAG, which means that each word is conditioned by its parent. For example, the probability of the sentence from Fig. 1 is factorized as follows: $P_{\mathtt{wp}}(s|T) = P(the|boy)P(boy|find)P(will|find)$ $P(find|\texttt{<NONE>})P(it|find)P(interesting|find)$.

3.2 Model wp,wg (Word Forms of Parent and Grandparent)

Inspired by [6], we also define model in which each word is conditioned by word forms of its parent and grandparent. For example, in phrase “listen to news” we have $P(news|\text{parent} = to, \text{grandparent} = listen)$.

3.3 Other Possible Models

Both the models, wp and wp,wg, are to be applied in *bottom-up factorization ordering*. Alternatively, we could define models in which a word is conditioned by its children, so the models would be applied in *top-down factorization ordering*. Similarly to [4], we could enhance the models with conditioning also on siblings.

[4] Charniak reports perplexity of trigram LM = 167, perplexity of trihead LM = 130, but the absolute values are not comparable to our results, because he uses a special “speech-like” corpus with reduced vocabulary, see [6] for details.

[5] Formally, $T = (V, r, \rho, ord)$, where V are nodes, $r \in V$ is the root, $\rho : V \setminus \{r\} \to V$ is a function which assigns parent nodes and $ord \subset V \times V$ is total ordering of nodes.

4 Additional Context Information

All LMs in our experiments (in Sect. 5) are computed using distribution of form $P(w|\mathbf{h})$, so they can be described by the context $\mathbf{h} = h_1, \ldots, h_F$ they use. Performance of LMs can be improved by supplying additional context factors h_f that can be used either for enlarging the context or for better smoothing. In addition to using various word-positions (e.g. preceding word, parent,...) in the context, we can extract various attributes from each word-position (word form, POS tag, number of children,...). Every context factor h_f has form $[attribute][word\text{-}position]$. For example, context tp,wp,tg means POS tag of parent, word form of parent and POS tag of grandparent. Possible attributes and word-positions are:

attributes	
w	word form
l	lemma
t	POS tag
T	coarse-grained POS tag
N	the word is Nth child of its parent
C	number of children
E	edge direction (left or right)

word-positions	
-1,-2,...	preceding words (1^{st}, 2^{nd},...)
+1,+2,...	following words (1^{st}, 2^{nd},...)
p	parent
g	grandparent
default	current word (applicable only for attributes N,C and E)

Attributes N and C are quantized – possible values are 0,1,2,3 and more.

5 Experiments

5.1 Data

For experiments, we used the data from CoNLL 2007 shared task [5], and we choose following seven languages: Arabic (ar), Catalan (ca), Czech (cs), English (en), Hungarian (hu), Italian (it), and Turkish (tr). Properties of this data are summarized in Tab. 1. The data are divided into two parts: train which was used for training the LMs, and test which was used solely for computing perplexity.

5.2 Four Experimental Settings

We consider four experimental settings which correspond to resources that may be available when using LMs in real applications:

Table 1. CoNLL data statistics. (OOV = percentage of test words not seen in train)

language	ar	ca	cs	en	hu	it	tr
sentences (train)	2,912	14,957	25,363	18,576	6,033	3,109	5,634
tokens (train)	111,669	430,844	432,296	446,573	131,799	71,199	65,182
unique words (train)	21,058	35,213	63,151	26,599	33,754	13,003	18,181
sentences (test)	130	166	285	213	389	248	299
tokens (test)	5,124	5,016	4,724	5,003	7,344	5,095	4,513
OOV	11.7%	3.8%	10.2%	2.6%	22.1%	12.0%	26.0%

- PLAIN: no additional information available, just word forms (Sect. 5.4),
- TAGS: part-of-speech (POS) tags and lemmata available (Sect. 5.5),
- DEP: topology of dependency trees available (Sect. 5.6),
- DEP+TAGS: topology, POS tags and lemmata available (Sect. 5.7).

In Sections 5.4 – 5.7 we compare perplexity of several LMs conforming the given setting. Note that the perplexity values are not directly comparable across the languages, because of different training data size and domain.

5.3 Smoothing

We use linear interpolation of models:[6]

$$P_{smoothed}(w|h_1,...,h_F) = \lambda_0 P_0(w) + \lambda_1 P_{ML}(w) + \sum_{f=1...F} \lambda_{f+1} P_{ML}(w|h_1,...,h_f),$$

where P_{ML} is the maximum likelihood estimate and P_0 is so-called probability of unseen words. $P_0(w) = 0$ if word w was observed in training data and $P_0(w) = 1/(dictionary\_upper\_bound - train\_vocabulary\_size)$ otherwise (we set $dictionary\_upper\_bound = 10^6$). Weights $\lambda_0...\lambda_F$ are trained using EM algorithm, so that they sum to one. We used 20% of the training data as “held-out data” for estimating the lambda weights. Note that the probability of unseen words always obtains the same weight λ_0 for all models – it is the ratio of held-out words that were not seen in the rest of training data.

5.4 PLAIN: Just Word Forms

As expected, enlarging the context lowers perplexity, i.e. bigrams have higher perplexity than trigrams, post-bigrams than post-trigrams, etc. However, for the training data sizes used in our experiments the improvement for 4-grams compared to trigrams is negligible.

Most surprising outcome of Tab. 2 is that post-ngram LMs have significantly better perplexity than standard ngram LMs. The difference is so prominent, that for five of the seven languages it is even better to use one following word as the context than two preceding words and for all the languages it is better to use two following words than three preceding words.

5.5 TAGS: POS Tags and Lemmata Available

In the TAGS setting (Tab. 3), we still do not exploit dependency structure of sentences – we try to lower the perplexity as much as possible just by enriching the context with additional attributes of following words.[7] The attributes are: POS tag (`t`), coarse-grained POS tag (`T`) and lemma (`l`). We trained a naïve tagger on the `train` data,

[6] It would be beneficial to compare also other smoothing techniques (see [7] for an overview), especially Generalized Parallel Backoff [2], but it is beyond the scope of this paper.

[7] We performed experiments also with additional attributes of preceding words, but similarly to the PLAIN setting, preceding words gave higher perplexity than following words. For clarity and space reasons, we do not show the results in Tab. 3.

Table 2. Perplexity of PLAIN models

Model	Perplexity						
	ar	ca	cs	en	hu	it	tr
w-1 (bigram)	2,052	368	3,632	387	5,203	1,606	4,034
w+1 (post-bigram)	2,006	337	3,391	355	4,735	1,440	3,720
w-1,w-2 (trigram)	1,988	325	3,530	356	5,183	1,552	4,015
w+1,w+2 (post-trigram)	1,950	301	3,298	328	4,721	1,399	3,712
w-1,w-2,w-3	1,989	324	3,531	355	5,183	1,553	4,015
w+1,w+2,w+3	1,951	299	3,299	327	4,721	1,400	3,712

which assigns the most frequent (full/coarse-grained) POS tag for a given word or the overall most frequent tag if the word was unseen in training data. Similarly, we created a naïve lemmatizer and tagged and lemmatized the `test` data.

We confirm the well-known finding (see e.g. [2]) that additional attributes improve the perplexity. In our experiments, for six of the seven languages it is even better to use one following word & its POS tag than two following words. In Tab. 3 we can see that adding coarse-grained POS tags to POS tags helps a little and adding lemmata helps only for some languages (it helps for morphologically rich languages such as Czech and Hungarian).[8]

Table 3. Perplexity of TAGS models

Model	Perplexity						
	ar	ca	cs	en	hu	it	tr
w-1,w-2 (trigram baseline)	1,988	325	3,530	356	5,183	1,552	4,015
t+1,w+1	1,706	310	2,999	316	4,091	1,277	3,346
t+1,w+1,t+2,w+2	1,641	276	2,909	287	4,067	1,246	3,340
T+1,t+1,w+1,T+2,t+2,w+2	1,641	271	2,901	286	4,059	1,243	3,315
T+1,t+1,l+1,w+1,T+2,t+2,l+2,w+2	1,611	271	2,884	286	3,924	1,242	3,060

5.6 DEP: Topology of Dependency Trees Available

We used Malt parser [8] and trained it on the `train` data with manual annotation of dependency structure, but automatic annotation of POS tags and lemmatization (using naïve tagger and lemmatizer from the previous section). Subsequently, we parsed both the `train` and `test` data. We trained and tested our dependency LMs on this new data.

Results in Tab. 4 show perplexity just for selected models (because possible configurations are numerous). Note that there is no single best ordering of context factors, e.g. for Arabic, Hungarian and Turkish it is better to use `N` before `wp`, while other languages have lower perplexity with the opposite ordering. The improvement of the DEP setting against PLAIN is even greater than with the TAGS setting.

[8] Actually, for English there are no lemmata in CoNLL 2007 data.

Table 4. Perplexity of DEP models

Model	Perplexity						
	ar	ca	cs	en	hu	it	tr
w-1,w-2 (trigram baseline)	1,988	325	3,530	356	5,183	1,552	4,015
wp	2,160	456	3,500	506	5,706	1,767	3,850
wp,wg	2,128	424	3,482	492	5,711	1,740	3,847
E,wp	1,993	327	3,070	391	4,891	1,393	3,386
E,C,N,wp	1,533	239	2,529	287	3,746	1,177	2,977
E,C,wp,N	1,570	232	2,410	277	3,879	1,137	2,988
E,C,N,wp,wg	1,525	229	2,522	283	3,746	1,174	2,976
E,C,wp,N,wg	1,561	222	2,403	272	3,879	1,135	2,986

5.7 DEP+TAGS: Both POS Tags and Topology Available

Tab. 5 shows that overall best results (i.e. lowest perplexity) for all the languages were reached by combination of both types of additional context information (DEP+TAGS).

Table 5. Perplexity of DEP+TAGS models

Model	Perplexity						
	ar	ca	cs	en	hu	it	tr
w-1,w-2 (trigram baseline)	1,988	325	3,530	356	5,183	1,552	4,015
E,C,Tp,tp,N,wp,Tg,tg,wg	1,303	202	2,077	242	3,392	1,051	2,659
E,C,Tp,tp,N,lp,wp,Tg,tg	1,280	209	2,034	244	3,346	1,052	2,555
E,C,Tp,tp,N,lp,wp,Tg,tg,lg	1,270	200	2,028	244	3,346	1,050	2,555

6 Discussion

Perplexity is traditionally considered as a good indication of LM performance in an intrinsic evaluation.[9] Fig. 2 illustrates that we achieved significant improvements in perplexity over the baseline (trigram LM) for all seven tested languages and all four settings (e.g. for English, PLAIN $\approx$ 8% improvement, TAGS $\approx$ 20% improvement, DEP $\approx$ 24% improvement, DEP+TAGS $\approx$ 31% improvement). We conclude with three main outcomes:

- In contrast to the common practice of using preceding words as context in language modeling, we observed better perplexity when using following words (i.e. post-ngram LMs).
- Dependency LMs achieved even better perplexity than post-ngram LMs.
- Using additional context information (e.g. POS tag and lemma) improved the perplexity both for post-ngram LMs and for dependency LMs.

[9] Of course, if we are interested in performance of a particular application, it is better to use extrinsic evaluation (e.g. WER for speech recognition and BLEU for MT).

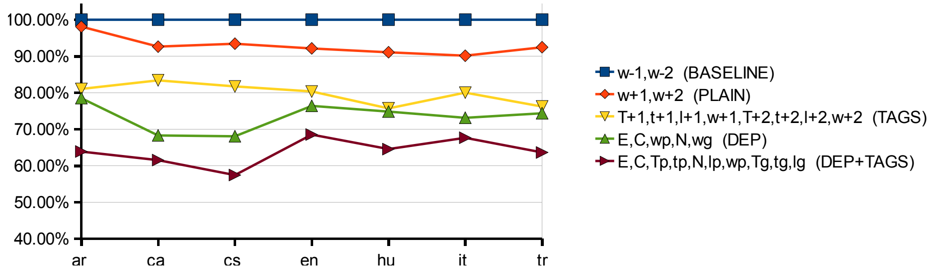


Fig. 2. Percentage comparison of perplexity of new LMs with the baseline (trigram LM = 100%). The best LM was chosen for each of settings: PLAIN, TAGS, DEP and DEP+TAGS.

The drawback of our dependency LMs is that they cannot be easily combined with ngram LMs nor translation/acoustic models at the word level. Unless we know the parse topology from another source, we must use a dependency parser that needs to see the whole sentence in advance, so the dependency LMs can be used merely for re-ranking of n-best lists of whole sentences. According to [6], this drawback of dependency LMs is not necessarily compelling. Similarly, we hope that the superior perplexity of our new LMs will result in improvements in real applications. In future, we would like to perform experiments with post-ngram LMs in speech recognition and with dependency LMs in tree-to-tree MT.

References

1. Rosenfeld, R.: A Maximum Entropy Approach to Adaptive Statistical Language Modelling. Computer speech and language 10, 187 (1996)
2. Bilmes, J., Kirchhoff, K.: Factored Language Models and Generalized Parallel Backoff. In: HLT/NAACL-2003, Edmonton, Alberta (2003)
3. Chelba, C., Engle, D., Jelinek, F., Jimenez, V., Khudanpur, S., Mangu, L., Printz, H., Ristad, E., Rosenfeld, R., Stolcke, A., et al.: et al.: Structure and Performance of a Dependency Language Model. In: Proceedings of Eurospeech (1997)
4. Shen, L., Xu, J., Weischedel, R.: A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model, pp. 577–585 (2008)
5. Nivre, J., Hall, J., Kubler, S., McDonald, R., Yuret, D., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 Shared Task on Dependency Parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, Prague, pp. 915–932 (2007)
6. Charniak, E.: Immediate-Head Parsing for Language Models. In: Processing of ACL 2001, vol. 39, pp. 116–123 (2001)
7. Chen, S., Goodman, J.: An Empirical Study of Smoothing Techniques for Language Modeling. Computer Speech and Language 13, 359–394 (1999)
8. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. Natural Language Engineering 13, 95–135 (2007)

Analysis of Czech Web 1T 5-Gram Corpus and Its Comparison with Czech National Corpus Data

Václav Procházka and Petr Pollák

Dept. of Circuit Theory, Czech Technical University,
Technická 2, 166 27 Prague, Czech Republic
prochva1@fel.cvut.cz, pollak@fel.cvut.cz
http://noel.feld.cvut.cz/speechlab/

Abstract. In this paper, newly issued Czech Web 1T 5-grams corpus created by Google and LDC is analysed and compared with reference n-gram corpus obtained from Czech National Corpus. Original 5-grams from both corpora were post-processed and statistical trigram language models of various vocabulary sizes and parameters were created. The comparison of various corpus statistics such as unique and total word and n-gram counts before and after post-processing is presented and discussed, especially with the focus on clearing Web 1T data from invalid tokens. The tools from HTK Toolkit were used for the evaluation and accuracy, OOV rates and perplexity were measured using sentence transcriptions from Czech SPEECON database.

Keywords: statistical language model, text corpora, Czech Web 1T 5-gram, Czech National Corpus, HTK Toolkit.

1 Introduction

The research in the field of Large Vocabulary Continuous Speech Recognition (LVCSR) has undergone intense development over the past few decades for many world languages, including languages and dialects spoken by rather small population, as e.g. Czech. Especially on the basis of increasing power of IT systems, more sophisticated speech technology applications can be currently seen in many systems used in daily human life. The first recognizers of singular commands or simple dialogue systems are joined by dictation machines converting voice input into written form, automated transcribers of audio-video records [1], moreover, on-line sub-titles generation in live TV broadcasts [2] was developed.

Current LVCSR systems typically use statistical language modelling, which describes in the simplest case the probability of single word occurrence in language (unigram) or in more general case appearance probability of n-word sequence (n-gram). Language modelling has significant importance for the accuracy of a constructed LVCSR system, and the increasing power of such systems is frequently based on the continuously increasing size of vocabulary and language model. It is very typical for Czech as well as for other languages of inflective nature that the vocabulary and language model must be very large to enable the recognition of natural continuous speech.

The creation of proper language models for Czech has also been done by other authors within the construction of the above described LVCSR systems. This article

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 181–188, 2010.

describes the newly publicly available Web based resource for language modelling in Czech [3] commonly with its comparison with reference Czech National Corpus data [4]. Using this data should prevent hard and time consuming work of collecting large text corpus for the generation of statistical n-gram model and thus it enables quicker realization of LVCSR experiments.

2 Analysed n-Grams Corpora

The Web 1T 5-gram corpora contain the set of n-grams obtained from Web sources by Google in cooperation with Linguistic Data Consortium [5]. The World Wide Web is frequently used as a source of various text corpora and the data from Web resources has usually dominant contribution within currently used text corpora in speech technology generally. Gathering text from such resource inherently means dealing with problems like collecting data only for the requested language, recognising and unifying encodings or filtering out markup tags. Consequently, assembling sufficient amount of texts for specific language is not straightforward. Crawling the Web and collecting text data is time consuming work and, finally, a high number of invalid words such as misspellings, other language words or random chunks of characters still appears in a collected text.

Web 1T 5-gram corpora offer already prepared sets of n-grams from the order of one up to the order of five. The first Web 1T 5-grams corpus was issued for English in 2006 [6] and within the last year this collection was extended by 10 other languages, including Czech [3].

2.1 Google Czech Web 1T n-Grams

The statistics of Czech Web 1T n-grams reports a very large amount of unique unigrams, almost 10 million. Initial analysis showed that it is due to the case sensitive manner of particular tokens and due to frequent appearance of semi-random character chunks as serial numbers, product codes, personal or company names, Internet nicknames, or punctuation marks. While contribution of punctuation marks to unique unigram count is insignificant, they are not necessary for creating basic bigram and trigram language models. It ultimately means that available n-grams must be filtered before further usage. As this corpus was built using texts which were rather variable and which were not carefully edited as a whole such as other collected corpora for language modelling purposes.

In the original corpus all tokens which were considered invalid (contains malformed, non-european or other invalid characters or are too long [3]) were replaced by the special token <UNK> (unknown word). Similarly all tokens with occurrence lower than 40 were also replaced by special token <UNK>. Subsequently all n-grams with occurrence count lower than 40 were discarded. Cutoff 40 is a common simplified notation for this operation. In addition to the original corpus, more tokens were mapped to special token <UNK>. These tokens do not form a proper Czech word, e.g. mixed strings of alphabetical and numeric characters or URLs. For this purpose the following Czech alphabet letters were considered valid: ’aábcčdďeéěfghiíjklmnňoópqrřsšťuúůvwxyýzž’.

Table 1. Additional class mappings done during 5-gram filtering of Czech Web 1T 5-gram and SYN2006PUB n-gram corpora

token content	token	example of string
alphanumeric characters and numbers	<UNK>	abc123
words with letter out of Czech alphabet	<UNK>	sjöberg
numbers 0–9 +-,.	<NUM>	123

Table 2. CNC SYN2006PUB n-gram corpus; unique n-gram counts

	original corpus	cleared corpus
unigram	9,786,424	1,804,682
trigrams	117,264,988	47,376,975
5-grams	103,280,138	51,747,413

Table 3. Czech Web 1T 5-gram corpus statistic; unique n-gram counts

	original corpus	after postprocessing
unigram	2,554,028	1,779,006
trigrams	189,152,100	170,243,851

During further processing new special token <NUM> was introduced to represent number expressions. Overview of all additional mappings introduced during post-procesing is presented in table 1. The meaning of special tokens for sentence start (<S>) and sentence end (</S>) was left unchanged. The count of unique unigrams in the corpus after this processing is approximately 1.8 million.

Discarding a token containing a punctuation mark from n-gram is not a straightforward operation. When such token is discarded, cross punctuation mark context is preserved but the order of n-gram is lowered. To preserve as much cross punctuation mark context as possible, this reduction was done using maximal available n-gram order, i.e. 5-grams reduced to the desired trigrams afterwards and n-grams of resulting order two or lower were discarded. Table 3 shows counts of selected n-gram orders before and after described processing.

Analysis performed on postprocessed trigrams also revealed that a lot of problematic tokens have remained in the corpus, e.g. foreign words (English, Slovak, German and other European languages) or variants of Czech words written without diacritical marks which is frequent practice to avoid problems with character encoding of Czech text over Internet. These tokens has been left untouched in the current level of the post-processing.

2.2 Czech National Corpus SYN2006PUB n-Grams

The above mentioned Web 1T 5-grams corpus was compared with the data obtained from Czech National Corpus [7] (CNC). CNC is a large corpus of written Czech collected within an academic project focused on the building and continuous extension of electronic resources of Czech texts and is supposed to be used as reference corpus in this work.

Limited access to this corpus is available via interactive Web, however, this interface does not provide the level of access needed for multigram generation. Multigrams used in this article as the reference for the analysis of Web 1T 5-grams corpus were obtained on the basis of bilateral agreement. We have the n-gram corpus generated from SYN2006PUB [4], i.e. a synchronic corpus of written journalism of 300 million of words (tokens). This corpus contains exclusively journalist texts from November 1989 to the end of 2004 which were not covered by other similar corpora SYN2000 and SYN2005 [7]. This n-gram corpus contains n-grams of the order one up to five, no cutoff for token or n-gram occurrence counts is used, and it also does not contain punctuation marks. Just one special token for the end of a sentence (</s>) is included. The number of unique unigrams in this corpus is approximately 2.5 million and tokens are case-sensitive. Unique n-gram counts are summarised in Table 2.

As punctuation marks are not present in this corpus, the usage of 5-grams does not bring any improvement in terms of preserving cross punctuation mark context, so trigrams were used for further analyses. Similary to processing Web 1T 5-grams, numeric expressions were also replaced by token <NUM>. In addition, end of sentence token </s> was expanded with pair of tokens for sentence start (<S>) and sentence end (</S>). Resulting 4-gram was splitted in two trigrams which were inserted back. It was done because the tools from HTK Toolkit [8] used in further steps require both these special tokens. Finally, these tokens were uppercased to keep them identical in both n-grams corpora.

3 Comparison of Analysed n-Gram Corpora

For the evaluation of a n-gram corpus various metrics may be used. Unique counts indicate how much diversified the original data are. Apart from comparing previously mentioned n-gram corpora to each other, basic comparison to Czech LC-StarII lexicon [9,10] is also shown.

3.1 Unigram Comparison

Firstly, the statistics of unique n-grams in particular corpora were computed and compared Statistics for SYN2006PUB n-gram corpus were counted with no cutoff, for Czech Web 1T 5-gram with original unchanged cutoff 40. Raw n-gram count in LC-StarII corpus corresponds to n-gram counts after postprocessing (cleaned) of other two n-gram corpora. Count of cleaned unigrams (proper names and abbreviations were deleted) is showed only for reference (Table 5). Results showed that the very high unigram counts in original Czech Web 1T n-gram corpus were reduced approximately 5 times by postprocessing. It yielded to the reduction of unique 5-gram counts to one half. The number of unigram in reference SYN2006PUB n-gram corpus was lowered by one quarter, the number of trigram decreased by 10%.

Comparison of unigram intersections between Web 1T 5-gram, SYN2006PUB 5-gram and LC-StarII corpora is shown in table 6. More than 1/3 of unigrams in Web 1T 5-gram corpus are also present in SYN2006PUB corpus. Almost all of the most common words represented by LC-StarII lexicon are found also in Web 1T 5-gram corpus. This result was expected as LC-StarII lexicon was created with the aim of covering 95% of words from representative Czech texts.

Similar comparison was also done for subsets of most frequent unigrams from Czech Web 1T 5-gram corpus and SYN2006PUB corpus. Subsets of size 60K, 120K,

Table 4. Intersections of unigram in Web 1T 5-gram and CNC SYN2006PUB 5-gram for most frequent unigrams in several limited vocabularies

	Web 1T 5-gram		SYN2006PUB 5-gram		LC-StarII	
	original	cleaned	original	cleaned	raw	filtered
unigram	9,786,424	1,804,682	2,554,028	1,799,005	132,574	84,724
trigrams	117,264,988	47,376,975	189,152,100	170,243,851	-	-
5-grams	103,280,138	51,747,413	302,836,997	302,770,408	-	-

Table 5. Summary of unique n-gram counts in original and cleared n-gram corpus

combination of corpora	count of common unigrams
CNC SYN2006PUB, Web 1T 5-gram, LC-StarII	83,856
CNC SYN2006PUB and Web 1T 5-gram	700,806
CNC SYN2006PUB and LC-StarII	84,587
LC-StarII and Web 1T	83,941

Table 6. Intersection of unigrams between analyzed corpora

vocabulary size	count of common unigrams
60,000	30,273
120,000	62,458
180,000	93,949
240,000	125,173

180K and 240K of most frequent unigrams were compared and counts of common unigrams are collected in Table 4. These subsets are later (Section 3.2) used also to limit vocabulary for language models.

3.2 Perplexity of Bigram and Trigram Models

Quality of language model (LM) is best measured by using LM together with the acoustic model in LVCSR and by measuring the achieved accuracy of recognized text. Another method, based on perplexity computation, quantifies LM quality without application in LVCSR. Perplexity is defined as $PP = 2^{LP}$ where

$$LP = -\frac{1}{N}\sum_{i=1}^{N}\log_2 q(x_i) \tag{1}$$

and it is often explained as mean log probability of each word for a piece q of previously unseen text of N pieces not used in building the language model. We use this measure for our first analysis, as it requires only LM and testing text corpus, however the perplexity does not necessarily tell exactly how well will analyzed LM perform in speech recognition.

For perplexity counting of created language models a subset of transcripts from Czech SPEECON [11,12] corpus was used as test data. The subset contains the total of 148,557 tokens in 14,914 sentences.

Table 7. Perplexity and n-gram count for language model trained from post-processed SYN2006PUB 3-gram corpus with cutoff 6

cut off 1			bigram model		trigram model	
vocabulary size	OOV rate	cut off	perplexity	bigram count	perplexity	trigram count
60,000	10.93%	1	932	17,587,483	928	65,783,406
120,000	6.36%	1	1,226	29,928,179	1,197	96,243,374
180,000	4.18%	1	1,432	35,987,677	1,400	108,421,863
240,000	3.06%	1	1,571	39,963,558	1,521	115,381,912

Table 8. Perplexity and n-gram count for language model trained from post-processed SYN2006PUB 3-gram corpus with cutoff 1

cut off 6			bigram model		trigram model	
vocabulary size	OOV rate	cut off	perplexity	bigram count	perplexity	trigram count
60,000	10.93%	6	939	4,156,790	817	6,733,685
120,000	6.36%	6	1,259	6,016,818	1,075	7,994,323
180,000	4.18%	6	1,483	6,739,430	1,263	8,289,070
240,000	3.06%	6	1,631	7,150,260	1,385	8,420,779

Table 9. Perplexity and n-gram count for language model trained from Czech Web 1T 5-gram corpus

			bigram model		trigram model	
vocabulary size	OOV rate	cut off	perplexity	bigram count	perplexity	trigram count
60,000	17.76%	40	49,507	11,437,940	162,509	35,166,960
120,000	12.14%	40	126,001	14,904,654	255,144	42,125,506
180,000	9.44%	40	153,682	16,522,260	664,866	43,882,665
240,000	7.63%	40	206,831	17,666,327	935,077	45,443,605

Currently, for corpus evaluation purpose, bigram and trigram models were created for several vocabulary sizes (60K, 120K, 180K, 240K). Models from SYN2006PUB 5-gram corpus were created with cutoff 1 and 6. Perplexities, out of vocabulary (OOV) rates and n-gram counts are summarised in tab. 8 and 7. The models from CNC SYN2006PUB show well known and expected pattern. Perplexities of these models are between 928 and 1,631. This is consistent with perplexity measurements from similar corpora presented in [14,15,16].

Exactly the same approach used for creation of models from SYN2006PUB n-gram corpus was used for the creation of models from Web 1T 5-grams corpus. These models were formally created with cutoff 1 (table 9) but effective cutoff is 40, according to cutoff of original corpus. Models created from this corpus have very high perplexity and significantly higher OOV counts than models from SYN2006PUB n-gram corpus (tables 8 and 7). There are several possible explanations and some of the reasons (foreign words, no diacritics in Czech words) have already been mentioned in Section 2.1. The nature of Web might be another reason, i.e. pages with parts of identical code (headers, footers, menus) or almost completely identical as Internet shops pages, where only small part of text might be different but the same common part will unproportionally increase the count of several particular n-grams. It means, that direct usage of these n-grams without further post-processing is not possible.

4 Conclusions

The analysis of recently issued and publicly available Czech Web 1T 5-grams corpus has been presented in this paper. The corpus was compared with the reference CNC SYN2006PUB 5-grams corpus in means of n-gram counts and perplexity computation on language models created from these corpora. The most important conclusions could be summarized as follows:

- The analyzed Web 1T 5-grams corpus seems to be a good source of data for language modelling. It offers large amounts of data from Web resources transformed into n-grams, with stripped (X)HTML markup, unified encoding and basic filtering. In spite of these advantages, it still contains a lot of unsuitable words so further filtering is necessary before the usage in LVCSR.
- Currently achieved perplexity with n-grams from cleaned Czech Web 1T 5-grams was still extremely high, i.e. more than 10^5 in comparison to results for reference SYN2006PUB n-grams corpus where low perplexity between 900 and 1,600 above chosen reference text corpus.

  It means that currently realized post-processing has not been sufficient yet and it should be extended by additional filtering to remove mainly words from foreign languages misspelled words, Czech words without accents or to decrease the influence of (almost) identical n-grams originating from identical page fragments, especially headers or footers.
- Next experiments will be performed with LVCSR using audio data from Czech SPEECON database to see whether the recognition accuracy will be correlated with achieved perplexity measurements.

Acknowledgements

The research was supported by the grant GAČR 102/08/0707 “Speech Recognition under Real-World Conditions” and by research activity MSM 6840770014 “Perspective Informative and Communications Technicalities Research”. The creation of LC-StarII Czech and Slovak lexica was supported by Harman/Becker, Ulm, Germany which is also the current owner of the collected lexica.

References

1. Nouza, J., Ždánský, J., David, P., Červa, P., Kolorenč, J., Nejedlová, D.: Fully Automated System for Czech Spoken Broadcast Transcription with Very Large (300K+) Lexicon. In: Interspeech 2005, Lisboa, Portugal (September 2005)
2. Pražák, A., Muller, L., Psutka, J.: LIVE TV SUBTITLING – Fast 2-pass LVCSR System for Online Subtitling. In: SIGMAP 2007. INSTICC PRESS, Lisabon (2007)

3. Brants, T., Franz, A.: Web 1T 5-gram, 10 European Languages, version 1. Linguistic Data Consortium, Philadelphia (2009), Web page http://www.ldc.upenn.edu
4. Czech National Corpus: Český národní korpus (Czech National Corpus) – SYN2006PUB. Institute of the Czech National Corpus FF UK, Praha (2006), http://www.korpus.cz
5. Linguistic Data Consortium: Home page (2010), http://www.ldc.upenn.edu
6. Brants, T., Franz, A.: Web 1T 5-gram, version 1. Linguistic Data Consortium, Philadelphia (2006), Web page http://www.ldc.upenn.edu
7. Czech National Corpus: Home page. Institute of the Czech National Corpus FF UK, Praha (2010), http://www.korpus.cz
8. Young, S., et al.: The Hidden Markov Model Toolkit (HTK), Version 3.4.1, Cambridge (2009), http://htk.eng.cam.ac.uk
9. Moreno, A.: LC-StarII. Lexica and Corpora for Speech-to-Speech Translation Components, http://www.lc-star.org
10. Pollák, P., Černocký, J., Smrž, P.: LC-STAR CSCZ. Czech lexicon for ASR and TTS (October 2008), http://www.lc-star.org
11. Pollák, P., Černocký, J.: Czech SPEECON Adult Database (November 2003), http://www.speechdat.org/speecon
12. Iskra, D., Grosskopf, B., Marasek, K., van den Heuvel, H., Diehl, F., Kiessling, A.: SPEECON – Speech Databases for Consumer Devices: Database Specification and Validation. In: Proc. of LREC 2002 (May 2002)
13. Young, S., et al.: The HTK Book, Version 3.4.1, Cambridge (2009), http://htk.eng.cam.ac.uk
14. Mikolov, T., Oparin, I., Glembek, O., Burget, L., Karafiát, M., Černocký, J.: Použití mluvených korpusů ve vývoji systému pro rozpoznávání přednášek (Use of spoken corpora in the development of system for recognition of Czéch lectures). In: Proc. of Čeština v mluveném korpusu (Czéch in Spoken Corpus), Praha (2007)
15. Mikolov, T.: Language Models for Automatic Speech Recognition of Czech Lectures. In: Proc. of STUDENT EEICT 2008, Brno (April 2008)
16. Byrne, W., Hajič, J., Ircing, P., Krbec, P., Psutka, J.: Morpheme Based Language Models for Speech Recognition in Czech. In: Proc. of Text, Speech, and Dialog 2000, Brno (2000)

Borda-Based Voting Schemes for Semantic Role Labeling

Vladimir Robles[1,2], Antonio Molina[2], and Paolo Rosso[2]

[1] Universidad Politécnica Salesiana, Cuenca, Calle Vieja 12-30, Ecuador
vrobles@ups.edu.ec
[2] ELiRF, DSIC, Universidad Politécnica de Valencia,
Valencia 46022, Camino de Vera s/n, Spain
{amolina,prosso}@dsic.upv.es

Abstract. In this article, we have studied the possibility of applying Borda and Fuzzy Borda voting schemes to combine semantic role labeling systems. To better select the correct semantic role, among those provided by different experts, we have introduced two measures: the first one calculates the overlap between labeled sentences, whereas the second one adds different scoring levels depending on the verbs that have been parsed.

Keywords: Semantic role labeling, Borda voting schemes.

1 Introduction

Previous studies shown that the semantic role labeling is a task that allows to improve the performance of many Natural Language Processing (NLP) applications. A semantic role is the underlying relationship between a syntactic constituent (consisting of a word or sequence of words) and the main verb of a sentence. The role is the function that assigns the predicate to its arguments. A clear example of what has been mentioned is shown in the following sentence: *"Hurricane-force winds demolished much of the town"*. If we review the sentence, it would have the following roles: *[Hurricane-force winds]*$_{\text{cause}}$ demolished *[much of the town]*$_{\text{theme}}$. The syntactic constituent *"Hurricane-force winds"* is the cause that leads to a certain event, while *"much of the town"* constituent represents the argument that undergoes a change of state. The main thematic roles are: agent (argument that produces the action), experiencer (argument that is subjected to a sensory, cognitive or emotional experience), container (argument that is good or bad in a situation), location (argument representing sites), action (argument expressing some dimension) and item (argument that undergoes a change of state).

The task of semantic role labeling has been studied from several machine learning approaches, including the use of probabilistic and statistical techniques, such as Maximum Entropy or Conditional Random Fields and methodologies based on artificial intelligence such as Support Vector Machines. These methodologies depend on choosing the relevant characteristics, representing information of various kinds: lexical, syntactic and probabilistic, among other types [1].

In this paper we review the possibility of applying Borda and Fuzzy Borda voting schemes [2], to determine the feasibility of combining various systems of semantic

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 189–196, 2010.

role labeling. To accomplish this task we have worked with the data set published in the shared task of the conference CoNLL 2005 (Conference on Computational Natural Language Learning)[1]. We worked with the corpus tagged by the 5 best systems. We defined two measures of analysis: the level of role overlapping and the role scoring tables contained in each sentence.

The rest of the paper is organized as follows. In Section 2 we review the features of the used corpus. The Borda voting scheme and its variant Fuzzy, and the possibility of using it to combine two or more role labeling systems are described in Section 3. In Section 4 we review the steps we used to combine the results generated by the CoNLL 2005 systems. In Section 5 we show the results and analyze them. Conclusion and future work are described in Section 6.

2 Corpus CoNLL 2005

In CoNLL 2005, the corpus used is based on Section 02 - 21 (training), Section 24 (development) and Section 23 (test) of the Wall Street Journal (WSJ). More precisely, the corpus is based on PropBank 1.0, which is a part of the Penn Treebank with enriched structures (predicate and argument). The corpus has different type of arguments, (i.e., Semantic Roles), Numbered Arguments (A0-A5, AA), Adjuntcs (AM-), References (R-), and Verbs (V) [3].

In Table 1 we can see a list of the characteristics of the 5 best systems. The systems are ordered by F-Measure. The table lists the name of participation of each system, as well as precision, recall and F-Measure.

Table 1. The best five systems from the CoNLL 2005 competition

System	Short Name	Precision	Recall	F-Measure
punyakanok	S_1	82.28%	76.78%	79.44
pradhan	S_2	82.95%	74.75%	78.63
haghighi	S_3	79.54%	77.39%	78.45
marquez	S_4	79.55%	76.45%	77.97
surdeanu	S_5	80.32%	72.95%	76.46

3 Borda Voting Schemes

The Borda voting schemes is a technique that has been used in several NLP tasks: word sense disambiguation [5], geographical information retrieval [4], named entity recognition [6]. In this context, we consider that this methodology can improve the performance of semantic role labeling by combining different systems. For example, in this sentence of the tWSJ corpus: *"As a result, the link between the futures and stock markets ripped apart."*, the best CoNLL 2005 three labeling systems produce the following results (Table 2):

[1] http://www.lsi.upc.edu/~srlconll/st05/st05.html

Table 2. Comparison of labeling process performed by the systems S_1, S_2 and S_3

Constituent	S_1	S_2	S_3
As	(AM-CAU*	(AM-CAU*	(AM-DIS*
a	*	*	*
result	*)	*)	*)
,	*	*	*
the	(A1*	(A1*	(A1*
link	*	*	*
between	*	*	*
the	*	*	*
futures	*	*	*
and	*	*	*
stock	*	*	*
markets	*)	*)	*)
ripped	(V*)	(V*)	(V*)
apart	(AM-DIR*)	(AM-MNR*)	(AM-DIR*)
.	*	*	*

If we want to apply a Borda voting scheme, each system should provide a determined amount of candidate roles for each sentence argument. In the example described in Table 3, the role AM-CAU is assigned to the argument *"As a result"*[2]. This argument must have been assigned to two or more candidate roles by each system. This allows the creation of the necessary matrices to apply the Borda voting scheme.

We calculate the general voting results considering role AM-CAU as candidate1, AM-LOC as candidate2 and AM-DIS as candidate3. For example, to calculate M_{S_1}, we fill with 1 in row 1 and column 2 which indicates that the system prefers candidate1 than candidate2. Doing so for all candidates and by filling 0 in the rest of positions, we obtain the matrix. The final vote is the sum of the rows of systems matrices.

$$M_{S_1} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad M_{S_2} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad M_{S_3} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad Final_{Vote} = \begin{bmatrix} 3 \\ 4 \\ 2 \end{bmatrix}$$

Table 3 shows the preference candidate order for each system and the general order of the Borda voting scheme.

Table 3. Order of preference after applying Borda voting scheme

S_1	S_2	S_3	Borda Preference
AM-CAU	AM-LOC	AM-DIS	**AM-LOC**
AM-LOC	AM-CAU	AM-LOC	**AM-CAU**
AM-DIS	AM-DIS	AM-CAU	**AM-DIS**

[2] To better illustrate our example, we add two candidate roles and we change the preference order for every role in the system S_2.

To apply a Fuzzy Borda voting scheme, we must add weights for each candidate role, as is shown in Table 4.

Table 4. Preference order for roles labeled by each system, using weights

S_1	S_2	S_3
AM-CAU: **8.2**	AM-LOC: **7.3**	AM-DIS: **9.2**
AM-LOC: **7.2**	AM-CAU: **5.2**	AM-LOC: **3.2**
AM-DIS: **6.7**	AM-DIS: **4.7**	AM-CAU: **2.7**

According to the Fuzzy Borda voting scheme, the element $r^i_{j,k}$ (row j, column k of the matrix M_{S_i} for the role labelling systems S_i) can be calculated using the following formula:

$$r^i_{j,k} = \frac{w^i_j}{w^i_j + w^i_k} \tag{1}$$

Using Formula 1, and the weights from Table 4, we calculate the preference matrix of Fuzzy Borda voting scheme:

$$M_{S_1} = \begin{bmatrix} 0.5 & 0.53 & 0.55 \\ 0.47 & 0.5 & 0.52 \\ 0.45 & 0.48 & 0.5 \end{bmatrix} \quad M_{S_2} = \begin{bmatrix} 0.5 & 0.42 & 0.53 \\ 0.58 & 0.5 & 0.6 \\ 0.47 & 0.39 & 0.5 \end{bmatrix} \quad M_{S_3} = \begin{bmatrix} 0.5 & 0.46 & 0.23 \\ 0.54 & 0.5 & 0.26 \\ 0.77 & 0.74 & 0.5 \end{bmatrix}$$

$$Final_{Vote} = \begin{bmatrix} 4.2 \\ 4.5 \\ 4.8 \end{bmatrix}$$

The resulting preference order, from the Fuzzy Borda scheme, is shown in Table 5.

Table 5. Preference order after applying Fuzzy Borda voting scheme

S_1	S_2	S_3	Preference Order
AM-CAU	AM-LOC	AM-DIS	**AM-DIS**
AM-LOC	AM-CAU	AM-LOC	**AM-LOC**
AM-DIS	AM-DIS	AM-CAU	**AM-CAU**

As we have seen, if we do not have the number of candidates or alternatives required by the Borda voting schemes, we can not apply them. We consider the following observations:

- In order to create the Borda matrix, each system must label roles as part of a single domain. All systems must assign the same candidate roles for each argument, ordered according to their preference or weights.
- The verb and its meaning are the parameters that help us to define candidate roles, which create the Borda matrix. Weights must be inferred from the level of precision, recall or F-Measure of a system.

4 Overlapping and Scored Verb Analysis

4.1 Overlapping

The level of overlap is a measure that allows us to analyze the level of matching between two or more role labeling systems. Therefore, high value of overlapping indicates that the criteria of these systems is closer. This allows us to select those two systems that have the greatest value of matching. The system that has the highest score of verb analysis is selected.

To illustrate how we calculate the overlapping we took the sentence "*'It screwed things up,' said one major specialist.*" from tWSJ corpus and the roles proposed by the systems S_1 and S_2 (Table 6).

Table 6. Sentence of corpus tWSJ labeled by the systems S_1 and S_2

	S_1		S_2	
Constituent	Verb screwed	Verb said	Verb screwed	Verb said
“	*	*	*	*
It	*	(A1*	(A0*)	*
screwed	(V*)	*	(V*)	(A1*
things	(A1*)	*	(A1*)	*
up	*	*)	*	*)
,	*	*	*	*
“	*	*	*	*
said	*	(V*)	*	(V*)
one	*	(A0*	*	(A0*
major	*	*	*	*
specialist	*	*)	*	*)
.	*	*	*	*
Verb Score	0.8225	0.8523	0.8	0.8

As shown in Table 6, for the analysis of the verb *"screwed"*, the system S_2 assigns A0 role to the constituent *"It"*, while the system S_1 does not. For the constituent *"things"*, both systems agree to assign A1 as role. In this case, there is an overlapping in a single argument. For the verb *"said"* there is a partial overlapping in A1 role, because for the system S_1 the argument is made up of *"It screwed things up"* constituents, whereas the system S_2 is made up of *"screwed things up"*. For the A0 role both systems assign the same constituents. In this case, there is a partial overlap (A1) and a full overlap (A0).

To calculate the overlaps that occur in arguments consisting of a single constituent, we assign a value of 1 and we add the other arguments that have a single constituent. For the verb *"screwed"*, the overlapping value 1.

In the case of partial overlapping, we consider how many overlapping constituents of an argument (CS) and how many constituents make up that argument (CF). To calculate this value we have derived the following formula:

$$Overlap = \sum_{1}^{N} \left(\frac{CS_{S1}}{CF_{S1}}\right) \cdot \left(\frac{CS_{S2}}{CF_{S2}}\right) . \tag{2}$$

Where:

- CS_{S1} and CS_{S2} represent the constituents that overlap in the labeled argument by the systems S_1 and S_2.
- CF_{S1} and CF_{S2} represent the constituents that make up the argument of the systems S_1 and S_2.
- N is the total number of roles in the sentence.

The level of overlapping for the verb *"said"* is calculated as follows:

$$Overlap = overlap_{A0-Role} + overlap_{A1-Role} .$$

$$Overlap = \left[\left(\frac{3}{3}\cdot\frac{3}{3}\right) + \left(\frac{3}{4}\cdot\frac{3}{3}\right)\right] = 1.75 .$$

4.2 Scored Verb Analysis

To calculate this value we use a scoring system for each labeled role in a sentence. The basis of this metric is the overall level of precision of each system for role labeling (recall and F-Measure could also be used). For example, the system S_1 labels A0 roles with a precision of 88.22%, recall of 87.88% and an F-Measure of 88.05.

Experiments were carried out using the precision values. By combining two or more role labeling systems, we are expanding the coverage level that the system has. To calculate the verb scores (Table 6), we obtain an average value of the precision that each system has to label the arguments of a specific verb. For example, in the system S_2, the score of the verb *"screwed"* is calculated as follows: [0.8(precision of labeling the role A0)+0.8 (precision of labeling the role A1)]/2=0.8.

For the verb *"screwed"*, the system S_2 has labeled two roles, while the system S_1 has labeled a single role. Our system selects the labeled roles of systems S_2 and S_1 for verbs *"screwed"* and *"said"*, respectively.

5 Experimental Results

In this section we show the main results that were obtained after applying our voting approach. We have used two schemes of overlapping and scoring, partial and complete overlapping. The first scheme does not discard the arguments that do not completely overlap. The second scheme discards those that do not have a complete overlapping (all its constituents, similar to a simple voting scheme). In Figure 1 we see the number of arguments correctly classified by each of the combinations that we have tested. The best result is achieved combining all the systems with partial overlapping. Figure 1 shows the roles that were misclassified and also those that the system was not able to classify. The combination that produces the best results, considering the two values together (roles misclassified and non-tagged), is S_1-S_3-S_4 with partial overlapping.

In Figure 2 we observe precision, recall and F-Measure for all system combinations. The combination that gets the best F-Measure value is S_1-S_3-S_4 with partial overlapping. The precision is affected by the number of systems involved in the combination, because not all the systems have optimal values of this measure.

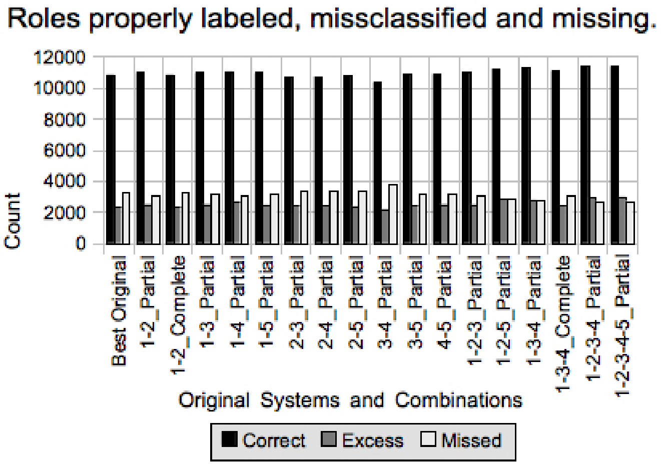


Fig. 1. Roles properly labeled, misclassified and missing

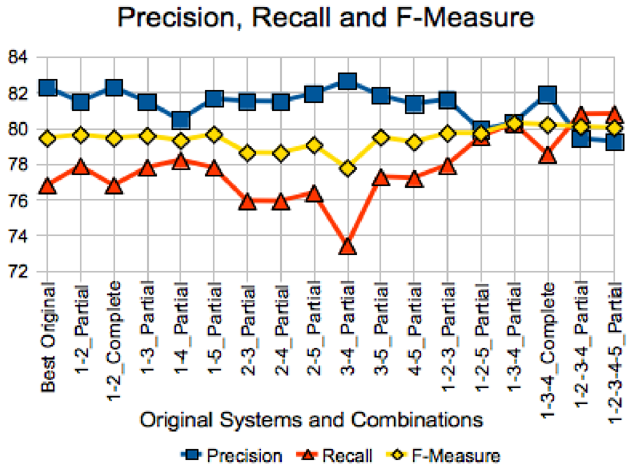


Fig. 2. Precision, recall and F-Measure of all system combinations

6 Conclusions

In this paper we have established an alternative measure of combinations between labeling systems, based on the Borda voting schemes. It has been shown that combining two or more systems together, better results can be achieved.

When we combine too many labeling systems, the precision become lower if these systems don not have similar values of precision. By contrast, the level of recall is enriched by the diversity of labeling schemes. One factor that improves the measurement of overlapping and especially the scored verb analysis, is to review the arguments that must have each verb. The implementation of this factor will help decrease the amount of roles that are misclassified or ignored.

As future work we propose to test scored verbs based on their level of matching with the arguments in PropBank and FrameNet, to apply Linear Integer Programming techniques which enrich the measurement process of overlapping and scored verb analysis, and include in the calculation of overlapping the values of precision, recall and F-Measure and verify their efficiency.

Acknowledgements

This work was partially funded by the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 and Telmosis PAID-06-08-3294 research projects.

References

1. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Computational Linguistics 28, 245–288 (2002)
2. García-Lapresta, J.L., Martínez-Panero, M.: Borda Count versus Approval Voting: A Fuzzy Approach. Public Choice 112, 167–184 (2002)
3. Carreras, X., Màrquez, L.: Introduction to the CoNLL 2005 Shared Task: Semantic Role Labeling. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, pp. 152–164 (2005)
4. Buscaldi, D., Perea, J.M., Rosso, P., Ureña, L.A., Ferrés, D., Rodríguez, H.: GeoTexMess: Result Fusion with Fuzzy Borda Ranking in Geographical Information Retrieval. In: CLEF, pp. 867–874 (2008); Revised Selected Papers
5. Buscaldi, D., Rosso, P.: UPV-WSD: Combining Different WSD Methods by Means of Fuzzy Borda Voting. In: Fourth International Workshop on Semantic Evaluations, pp. 434–437 (2007)
6. Benajiba, Y., Diab, M., Rosso, P.: Arabic Named Entity Recognition using Optimized Features Sets, EMNLP, Hawaii, USA (2008)

Towards a Bank of Constituent Parse Trees for Polish

Marek Świdziński[1] and Marcin Woliński[2]

[1] Institute of Polish, Warsaw University
[2] Institute of Computer Science, Polish Academy of Sciences

Abstract. We present a project aimed at construction of a bank of constituent parse trees for 20,000 Polish sentences taken from the balanced hand-annotated subcorpus of the National Corpus of Polish (NKJP).

The treebank is to be obtained by automatic parsing and manual disambiguation of resulting trees. The grammar applied by the project is a new version of Świdziński's formal definition of Polish. Each sentence is disambiguated independently by two linguists and, if needed, adjudicated by a supervisor. The feedback from this process is used to iteratively improve the grammar.

In the paper, we describe linguistic but also technical decisions made in the project. We discuss the overall shape of the parse trees including the extent of encoded grammatical information. We also delve into the problem of syntactic disambiguation as a challenge for our job.

Keywords: treebank of Polish, constituent parse trees, DCG grammar.

1 The Project

This paper reports on a project aimed at building a treebank of Polish.[1] To the best of our knowledge, it will be the first large treebank of Polish. In the present project trees for about 20,000 sentences will be compiled. The project will finish in a year but a follow-up work is already planned.

The treebank is built in a semi-automatic process. Parse trees (or, rather, parse forests) are generated by a parser and then selected and validated by human annotators. A linguist tries to choose the right tree for each sentence. If there is no valid tree in the forest the linguist can judge the sentence ungrammatical; in this case the processing ends. However, if the sentence is judged correct it is passed to the authors of the grammar, who will improve/correct the grammar, and a new parse forest will be presented for reconsideration.

Thus, the process is iterative: the grammar and the treebank are developed in parallel, as advocated for e.g. by [1,2]. Note, however, that since we work with a constituent grammar we cannot draw much experience from other projects for Slavic languages, as they work mostly in dependency formalisms (most notably the PDT [3]).

The process is facilitated by a web-based environment Dendrarium [4] developed especially for our treebank. Every sentence is processed independently by two users.

[1] The project is partially funded by the research grant N N104 224735 from the Polish Ministry of Science and Higher Education.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 197–204, 2010.

Then, if their answers differ a supervisor makes a choice. An important feature of the system is that it allows for avoiding repeated work caused by changes in the grammar.

We work on one-million word balanced subcorpus of the National Corpus of Polish (NKJP, `http://nkjp.pl`, [5,6]). The subcorpus has been manually annotated with morphological features, which means that the output of a morphological analyser is disambiguated and grammatical features of words unknown to the analyser (mainly proper names) are added. Consequently, every word in the subcorpus bears exactly one morphological interpretation.

In the current project we have decided not to correct the texts or their morphological interpretations we receive from the NKJP project. Such corrections may be applied in a follow-up work.

Moreover, we do not assume that we are able to process every correct Polish utterance. We decide to describe only finite sentences (i.e., those that are based upon the finite verb) and coordinate sentences, leaving other types of expressions untouched. As we use a constituent-based formalism discontinuous structures pose a problem. In the present project we will cope with but a very limited set of types of such phrases, which means that most of discontinuous sentences will not get parse trees.

What follows from all this is that our treebank will be biased by the grammar. However, due to limited resources we are not able to achieve a 100% coverage on the corpus. Therefore, we will classify utterances rejected by the parser, which will hopefully allow us to reduce the bias in the follow-up work.

2 The Grammar

The grammar used in the project is a new version of Marek Świdziński's grammar of Polish [7] expressed in the Definite Clause Grammar formalism [8] and implemented as the Świgra parser [9]. The grammar has undergone a deep reconstruction. The set of nonterminals has been limited (e.g., seven clause types have been reduced to one), resulting in trees that are much simpler, their height being significantly reduced. We are in the process of supplementing the grammar with rules that define constructions neglected in the previous version [10].

The new grammar redefines finite sentences. The whole definition is a set of three rules that account for all possible permutations of three types of components: finite phrase (verb) plus 0–3 required phrases (arguments) and 0–3 free phrases (adjuncts)[2]. The DCG apparatus has been extended to allow right hand sides of rules to include sequences of nonterminals of arbitrary length.

The grammar accounts for punctuation as a sophisticated syntactic phenomenon. Commas are strongly context-bounded in Polish. To cope with the problem special parameters are introduced in most rules that either block comma appearance in some contexts or force it in other.

Another interesting feature is that rules provide information which of the components of the construction defined by a given rule is its center (head). It will make it possible to generate dependency trees from constituent ones (cf. [11]).

[2] There are three other auxiliary compoments not worth mentioning here.

It should be emphasized that the Świdziński's grammar has an ambition to catch all possible structurizations (interpretations) for sentences examined. As the parser does not include any statistical component it is typical that we get large sets of trees even for short (i.e., simple) sentences. Therefore the problem of syntactic disambiguation is of crucial importance in the project.

3 The Structure of Parse Trees

Figure 1 presents an example of a parse tree, as displayed in our treebanking environment.

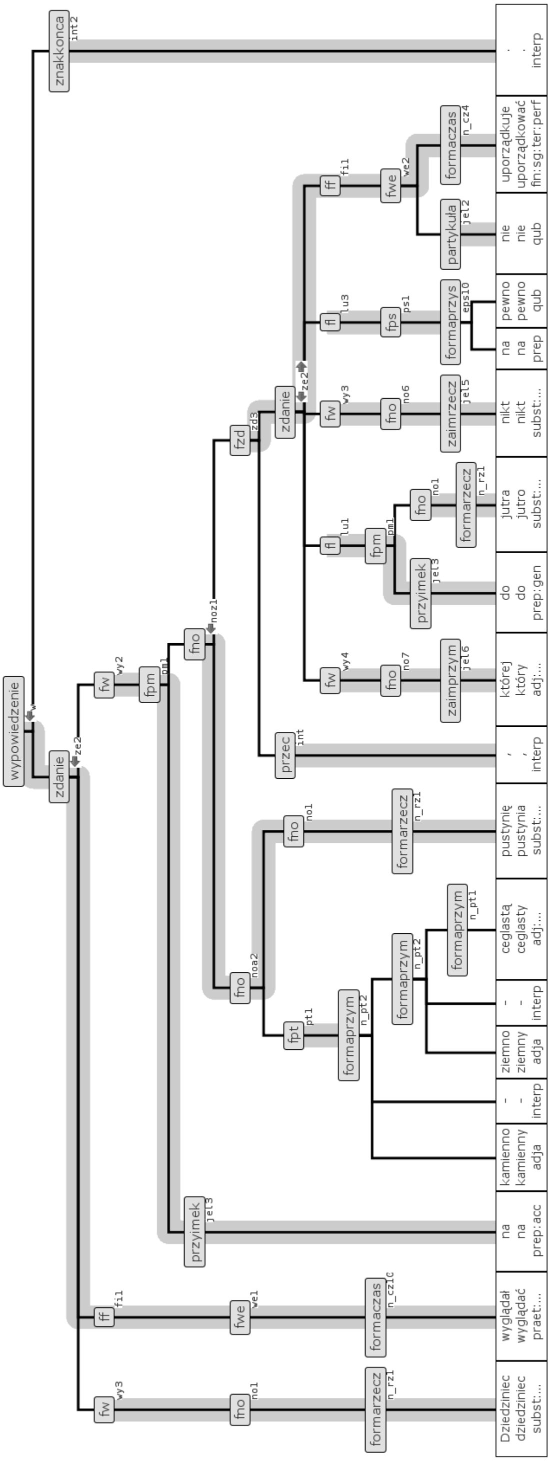
As mentioned before, the text analysed comes from the National Corpus of Polish, and both tokens and their morphological description originate directly therefrom. Tokens are displayed as a series of boxes at the bottom of the tree. Each box contains a word-form, a lemma, and morphological characteristics in the IPI PAN Tagset notation [12,13].

Nodes of the tree are assigned names of nonterminal units. Thick gray shadows emphasising some branches in the tree show distributional centres of phrases. They join each of the nonterminals with the token representing its centre. This way the utterance (wypowiedzenie) and the main clause (zdanie) in the example are connected with the verb *wyglądać* which is the centre of the whole sentence. For nominal phrases the subordinating noun is taken for their centre. Exocentric constructions get an arbitary interpretation. For example, prepositional-nominal phrases fpm have the preposition as their centre.

Nonterminal nodes in the tree can be seen as organised into several layers of annotation starting from the bottom.

The first layer represents what we call 'syntactic words' (the term coined by Przepiórkowski [14]). It comprises units like formarzecz (nominal form), zaimrzecz (nominal pronoun), formaczas (verbal form), przyimek (preposition), formaprzym (adjectival form), zaimprzym (adjectival pronoun), formaprzys (adverbial form), partykuła (particle), przec (comma). Typically, each corresponds to just one token. However, tokens (or segments) in NKJP are rather fine-grained, and sometimes a sequence of tokens (words) is taken for one unit on the 'presyntactic' level. For example, past forms of verbs are, following NKJP, formalised as composed of a 'past participle' and an agglutinated auxiliary form of the verb 'to be' (e.g., the form *zrobiłem* 'I-did' is a sequence of *zrobił-*, which carries tense and gender, and the auxiliary *-em* carrying person and number). The two segments need not be adjacent; the latter can precede the former, attached to some word before (sometimes obligatorily), as in *Chcesz, żebym to zrobił.* lit. you-want that-I-masc-sg it do 'You want me to do it.' (cf. [15]).

A similar approach applies to compounded adjectives—expressions locating at the boundary between lexicon and syntax. In Polish, we have got a special compound-building form of adjectives that ends in *-o* (marked adja in the IPI PAN Tagset) to be prefixed to the regular forms of adjectives with or without use of hyphen, e.g., *biało-czerwony* 'white and red' (as the Polish flag), *żółtozielony* 'greenish yelow', or *polsko-ukraińsko-rosyjski* 'Polish Ukrainian Russian' (e.g., a summit). This type of compounding is recursive. Similar nominal expressions exist in Polish as well though compounding is fairly more restricted.



Dziedziniec wyglądał na kamienno-ziemno-ceglastą pustynię, której do jutra na pewno nikt nie uporządkuje.
courtyard looked on stony earthy bricky desert which until tomorrow on sure nobody not will-tidy-up

'The courtyard looked like a stony, earthy, and bricky desert that no one will for sure tidy up until tomorrow.'

Fig. 1. An example of a parse tree. Morphological descriptions have been shortened. For explanation of particular labels see text.

Moreover, syntactic words are used to cater for analytical (multi-token) verb forms (e.g., *będzie robić* 'will do'), reflexive verbs (e.g., *boi się* lit. 'is-afraid-of self'), and some idiomatic multiword expressions (e.g., *na pewno* 'surely'). The last expression is treated as an adverb in the tree, while its tokens being the preposition *na* and an idiosyncratic adjectival word form *pewno* that appears only in this idiom (cf. [16]).

Although represented with regular nodes and edges, we treat this level as pre-syntactic since the units in question differ from standard syntactic units in some important ways. In particular, it is rather hard to identify their centres. We assume that the whole syntactic word is the centre for higher-level structures. It is also impossible to substitute higher level structures, like the adjectival phrase, for components thereof (e.g., **biało-ciemno czerwony* 'white and dark red' is not possible), which in itself justifies introducing special units for them.

The second layer of nonterminal units represents constituent phrases understood morphologically: verbal phrases fwe, nominal phrases fno, adjectival phrases fpt, prepositional-nominal phrases fpm, "sentential" phrases (i.e., subordinate clauses) fzd, and so on. Obviously, we allow for an arbitrary level of complication within constituents. They may be coordinate structures, include modifiers of various types, contain embedded clauses, etc. (cf. [10])

The third layer is needed to reveal clause structure, according to the definition of sentence (zdanie) in Świdziński´s grammar. The phrases of the second level are classified according to their function as constituents of a clause. On this level we locate the finite phrase ff, which is the clause centre, and its dependents: required phrases (arguments) fw and free phrases (adjuncts) fl. Subject of the clause is one of its required phrases, others being complements. This way valence frames are easily visible in the tree structure.

The fourth layer comprises clauses. Simple clauses consist of phrases of the third level. Subordinate clauses are regarded as phrases in our account. There are also coordinate clauses (not present in the example sentence).

Finally, the root of the tree is utterance (wypowiedzenie). It consists of a clause and a final punctuation (znakkonca). As can be seen, punctuation characters are treated as constituents in the tree.

The Świdziński's grammar assigns numerous attributes to nonterminals; we omit them in Fig. 1 to save space. They include morphological features but most attributes are, in fact, purely syntactic. They formalise various contextual co-occurrence restrictions. We have got, e.g., a parameter responsible for clasification of syntactic units (clauses and phrases) according to whether they contain an interrogative or relative terminal or whether they can serve as, or appear within, a subordinate clause of a given type. The general philosophy of the grammar is that everything is bounded (or controlled in a non-Chomskyan sense) in Polish expressions ("overagreement" philosophy).

4 Syntactic Ambiguities

Since we work on a manually disambiguated subcorpus of NKJP we are spared the issue of morphological ambiguities. It means that ambiguities we have to deal with are of structural or syntactic nature.

In the project disambiguation of parse forests is performed in terms of ambiguous nodes, i.e., nodes where different subtrees spanning the same tokens can be attached (cf.[4]). This happens when several rules can alternatively be used to obtain the same nonterminal with a fixed set of attributes.

According to our experiences so far, the node for clause (zdanie) seems most difficult. Variant realisations for a clause can be quite numerous (over 50) due to interaction of several problems:

1. Some constituents can be split into smaller units or aggregated further on. For example, in the relative clause taken from our example we find the free phrase *do jutra* and the required phrase *nikt* as components. They could be combined into one phrase. The phrase would not have any reasonable semantic interpretation but from the purely syntactic point of view it is perfectly admissible, since the phrase *do domu droga* 'way home' with the same structure is plausible. This type of variation leads to different number and extents of constituents of the clause.

2. The problem of distinguishing complements from adjuncts provides another source of ambiguity. In our formalism the distinction results in variants of zdanie differing only in labelling particular components as a required phrase fw or a free phrase fl. This ambiguity can be limited to some extent by appropriate entries in the valence dictionary. Unfortunately, it cannot be completely avoided. Most of possible realisations of required phrases are shared by free phrases. Even if we get a complement interpretation for a given component on the basis of valence data the parser will give an adjunct interpretation as well (we do not demand that valence frames be realised "non-elliptically").

A simple example can illustrate the point. Nominal phrases in accusative serve as a very common complement. Some accusative phrases can also be adjuncts describing the length of events (*godzinę* '[for] an hour', *chwilę* '[for] a moment', *dwie kolejki* '[for] two rounds'). Neither rejection of accusative nominal realization of fl by the grammar, nor allowing for it on a lexical basis (a list of possible centres of such fl's) seems easy to do. The decision has to be taken by annotators.

3. An additional level of complication pertains to verbs that require adverbial phrases. Required adverbial phrases can often be substituted with prepositional-nominal phrases with various prepositions. For example, verbs of location, movement or translocation, like *mieszkać* 'live', *pojechać* 'go (to)', or *zanieść* 'carry sth somewhere' require adverbial phrases. For each of them the adverbial phrase is substitutable by a prepositional-nominal one; cf., e.g., *Mieszkam tutaj (w Warszawie, nad morzem, pod Toruniem, ...).* 'I live here (in Warsaw, at the seaside, close to Toruń, ...)'. The parser of course allows for such substitutions, and annotators have to decide whether a given required phrase is prepositional or adverbial. The problem is limited to verbs missing from the valence dictionary. In such cases the parser uses a default frame allowing both for advp and prepnp.

In other ambiguous nodes of our trees variants are generally much less numerous. The main problem here is that alternative modifier attachments lead to various chunkings of a phrase. So, the number of variants in a particular node is generally limited by the number of spanned tokens; usually it is 2 or 3. The worst case of ambiguity is represented by so called genitive clusters, i.e., series of nouns in genitive, each of which can modify any other (respecting continuity). We have seen such sequences of the length of 11; theoretically, the length is unlimited.

Let us illustrate it with an example. The nominal phrase *ostatnie słowa Stalina* 'the last Stalin's words' can be structured twofold: either as *(ostatnie) (słowa Stalina)*, or as *(ostatnie słowa) (Stalina)*. Both interpretations are fully correct. No semantic, thematic-rhematic, or pragmatic difference takes place here.

All this shows that human disambiguators are unavoidable in the project. One can assume that each utterance, regarded as a part of a given speech act, meets exactly one interpretation. Therefore, our experts are provided with an (informal) instruction of how to evaluate outputs of the parser. Actually, the instruction gives room to semantic reflection and, moreover, access to the context.

There are some semi-formal (superficial) prompts to include in the instruction for our disambiguators. If a given prepositional-nominal phrase is a constituent of the idiomatic verbal expression it is a complement, not adjunct (*umrzeć ze śmiechu* 'die of laughter'). If such a phrase must be asked about with this preposition it is a prepositional, not adverbial complement (*pojechać do Piotra* 'go to Peter': *do kogo?* 'to whom?', and not *dokąd?* 'where…to?').

5 Conclusions

Probably the most important message of this paper is that finally, after years of Polish lagging behind, a relatively large treebank of Polish is under construction. Since there are no previous experiences with large treebanks of Polish we consider the present project a pilot work, in which we will recognise the main obstacles to be taken up later. In particular we assume to build a catalogue of types of discontinuous phrases in Polish, which will need some special treatment.

A follow-up project is already scheduled, which, we hope, will allow us to include sentences skipped in the pilot phase and to enlarge the treebank to 50,000 sentences in 3 years time.

Annotation of the treebank is in progress, so we are yet unable to provide any evaluation.

We assume that the treebank we wish to obtain will allow us to reformulate the grammar so that it could block some spurious interpretations. This pertains in particular to the problem of free phrases. We already see two possible directions of development for our grammar, both based on the treebank: lexicalisation and including some statistical component.

References

1. Branco, A.: LogicalFormBanks, the Next Generation of Semantically Annotated Corpora: Key Issues in Construction Methodology. In: Kłopotek, M.A., et al. (eds.) Recent Advances in Intelligent Information Systems, Exit, Warsaw, pp. 3–11 (2009)
2. Rosén, V., de Smedt, K., Meurer, P.: Towards a Toolkit Linking Treebanking to Grammar Development. In: Hajič, J., Nivre, J. (eds.) Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories, pp. 55–66 (2006)

3. Böhmová, A., Hajič, J., Hajičová, E., Hladká, B.: The Prague Dependency Treebank: A 3-level Annotation Scenario. In: Abeillé, A. (ed.) Treebanks. Building and Using Parsed Corpora, pp. 103–127. Kluwer Academic Publishers, Dordrecht (2003)
4. Woliński, M.: Dendrarium – an Open Source Tool for Treebank Building. In: Kłopotek, M.A., et al. (eds.) Intelligent Information Systems, Siedlce, pp. 193–204 (2010)
5. Przepiórkowski, A., Górski, R.L., Łaziński, M., Pęzik, P.: Recent Developments in the National Corpus of Polish. In: Proc. of LREC 2010, ELRA (2010)
6. Przepiórkowski, A., Górski, R.L., Lewandowska-Tomaszczyk, B., Łaziński, M.: Towards the National Corpus of Polish. In: Proc. of LREC, ELRA (2008)
7. Świdziński, M.: Gramatyka formalna języka polskiego. Rozprawy Uniwersytetu Warszawskiego. Wydawnictwa Uniwersytetu Warszawskiego, Warszawa (1992)
8. Pereira, F., Warren, D.H.D.: Definite Clause Grammars for Language Analysis – a Survey of the Formalism and a Comparison with Augmented Transition Networks. Artificial Intelligence 13, 231–278 (1980)
9. Woliński, M.: Komputerowa weryfikacja gramatyki Świdzińskiego. Ph.D. thesis, Instytut Podstaw Informatyki PAN, Warszawa (December 2004)
10. Świdziński, M., Woliński, M.: A New Formal Definition of Polish Nominal Phrases. In: Aspects of Natural Language Processing. LNCS, vol. 5070, pp. 143–162. Springer, Heidelberg (2009)
11. Nivre, J.: Theory-Supporting Treebanks. In: Proceedings of the Second Workshop on Treebanks and Linguistic Theories (2003)
12. Przepiórkowski, A.: A Comparison of Two Morphosyntactic Tagsets of Polish. In: Koseska-Toszewa, V., Dimitrova, L., Roszko, R. (eds.) Representing Semantics in Digital Lexicography, Warsaw, pp. 138–144 (2009)
13. Przepiórkowski, A., Woliński, M.: A Flexemic Tagset for Polish. In: Proceedings of the Workshop on Morphological Processing of Slavic Languages, EACL 2003, pp. 33–40 (2003)
14. Przepiórkowski, A.: Powierzchniowe przetwarzanie języka polskiego. Exit, Warsaw (2008)
15. Przepiórkowski, A., Woliński, M.: The Unbearable Lightness of Tagging: A Case Study in Morphosyntactic Tagging of Polish. In: Proc. of the 4th Workshop on Linguistically Interpreted Corpora (LINC 2003), EACL 2003, pp. 109–116 (2003)
16. Derwojedowa, M., Rudolf, M.: Czy burkina to dziewczyna i co o tym sądzą ich królewskie mości, czyli o jednostkach leksykalnych pewnego typu. Poradnik Językowy 3 (2003)

Coverage-Based Methods for Distributional Stopword Selection in Text Segmentation

Joe Vasak and Fei Song

Department of Computing and Information Science, University of Guelph
Guelph, Ontario, Canada N1G 2W1
{jvasak,fsong}@uoguelph.ca

Abstract. Unlike the common stopwords in information retrieval, distributional stopwords are document-specific and refer to the words that are more or less evenly distributed across a document. Isolating distributional stopwords has been shown to be useful for text segmentation, since it helps improve the representation of a segment by reducing the overlapped words between neighboring segments. In this paper, we propose three new measures for distributional stopword selection and expand the notion of distributional stopwords from the document level to a topic level. Two of our new measures are based on the distributional coverage of a word and the other one is extended from an existing measure called distribution difference by relying on the density of words in a way similar to another measure called distribution significance. Our experiments show that these new measures are not only efficient to compute, but also more accurate than or comparable to the existing measures for distributional stopword selection and that distributional stopword selection at a topic level is more accurate than document level selection for subtopic segmentation.

1 Introduction

Text segmentation divides a document into a sequence of segments that roughly correspond to topics. In general, a topic should be cohesive in that there are connections between consecutive sentences within a region of the text, usually by means of word repetitions (called lexical cohesion). Most text segmentation approaches use lexical cohesion to measure connectivity between topics so that we can break text at points where the connectivity between neighboring segments is the weakest [1,2,3]. Many advanced applications benefit from knowing the segment structure of a document, such as information retrieval, question answering, text summarization, text classification, and information visualization.

However, these cohesion-based approaches are not adequate for many naturally occurring documents which we call coherent documents. In such a document, information is typically organized into a hierarchical structure where there are a set of interrelated topics that contribute to one or few common themes and some topics may be further supported by subtopics to provide details. In cohesive documents such as news feed, topics are often unrelated and transitions from one to another are relatively clear [2]. However, in coherent documents such as expository text, topic transitions are subtler and more difficult to detect [1,4] given the hierarchical organization with interrelated topics contributing to one or few common themes.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 205–215, 2010.

In this paper, we try to enhance the separation between adjacent topics in coherent documents by identifying words that are more or less evenly distributed across a document, called distributional stopwords. In particular, we propose three new measures for distributional stopword selection: two are based on the distribution coverage of a word (i.e., the maximum span of a word across the sentences in a document) and the other is obtained from combining two existing measures [5]. In addition, we expand the notion of distributional stopwords from the document level to a topic level so that we can improve the performance of subtopic segmentation for coherent documents. We also conduct experiments on coherent text to demonstrate the effectiveness of these new measures for text segmentation and the usefulness of regional distributional stopword selection for subtopic segmentation.

2 Previous Work on Distributional Stopwords

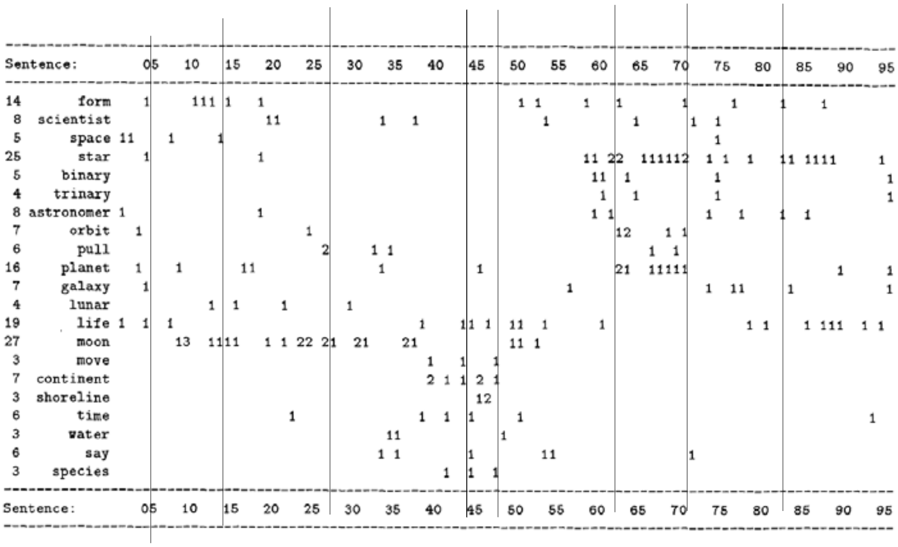
Lexical cohesion refers to the connectivity between two words in terms of word relationships [6]. Although there can be many kinds of relations between words, the simplest form is identical word repetition. The relationship links between words capture the distribution of a word through repetitions and are often modeled by lexical chaining, which keeps track of all occurrences of a word's position throughout a document. Lexical chaining is usually represented by word repetitions at the sentence level [7,5], where the distribution includes all the sentence occurrences of a word in the text. Figure 1 shows the distributions of selected words from the Stargazer text [1].

Using the distribution of a word across all sentences in text, we can identify distributional stopwords, since they are more or less evenly distributed across the text. This type of stopword is similar to common stopwords in information retrieval in that they do not help distinguish one segment for another given that they are more or less evenly positioned and appear in many different segments. However, unlike common stopwords that occur frequently in text (such as “the”, “of”, and so on), distributional stopwords are document-specific and can appear at all frequency levels.

As shown in Figure 1, words like “form”, “scientist”, and “say” can be treated as distributional stopwords since they are distributed somewhat evenly in the text and appear in many segments. On the other hand, words like “shoreline” and “species” are highly concentrated in a small region of the text and thus are good indicators for text segmentation. As a result, we can call such words distributional content words.

To capture distributional stopwords in text, we need a measure that can separate the content words from the stopwords. Ji and Zha computes the distribution variance of a word over multiple partition levels [4]. A document is first partitioned evenly into two segments, then three segments, and so on. At each level, word frequencies are used to calculate the variance of a word against the average word frequency. By averaging the variances across all partition levels, we get a measure for selecting distributional stopwords: words with the average variance below a pre-specified threshold are considered evenly distributed and thus can be removed for text segmentation.

Vasak and Song introduced two different measures for distributional stopword removal: distribution difference and distribution significance [5]. Distribution difference is aimed at improving Ji and Zha's multi-level partition method, since computing word



```
-------------------------------------------------------------------------------------------------------------
Sentence:        05   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90   95
-------------------------------------------------------------------------------------------------------------
14       form    1     111 1   1                                   1 1     1    1       1     1     1    1
 8  scientist                  11                  1   1                   1          1      1  1
 5      space 11    1     1                                                            1
25       star    1             1                                          11 22  111112  1 1  1    11 1111     1
 5     binary                                                             11   1          1                    1
 4    trinary                                                              1   1          1                    1
 8 astronomer 1                1                                          1 1         1   1    1  1
 7      orbit   1                     1                                      12    1 1
 6       pull                         2     1 1                                  1  1
16     planet   1    1        11               1           1                 21  11111                  1     1
 7     galaxy    1                                                   1                  1  11     1            1
 4      lunar            1  1     1       1
19       life 1  1  1                                1     11 1  11  1      1                 1 1    1 111  1 1
27       moon        13   1111   1 1 22 21  21    21           11 1
 3       move                                             1   1   1
 7  continent                                             2 1 1 2 1
 3  shoreline                                                   12
 6       time                       1                    1  1     1     1                                   1
 3      water                                       11          1
 6        say                                     1 1         1        11                1
 3    species                                               1  1  1
-------------------------------------------------------------------------------------------------------------
Sentence:        05   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90   95
-------------------------------------------------------------------------------------------------------------
```

Fig. 1. Distribution of selected terms from the Stargazer text, with a single digit frequency per sentence number (vertical lines indicate segment boundaries and blanks indicate a frequency of zero); (Taken from Hearst 1994)

variances over multiple levels is computationally expensive and at the coarse/low partition levels the variances tend to be close to the average word frequencies. By identifying one suitable partition level for each word, both efficiency and accuracy can be improved with the distribution difference method. Distribution signification is based on the density of a word, which is inversely proportional to the distance between two adjacent occurrences of the same word. Intuitively, highly concentrated words will have multiple occurrences close to each other and are considered usefull for text segmentation.

3 Regional Distributional Stopwords

Work by [4,5] shows that distributional stopword removal is particularly useful for the segmentation of coherent documents where information is hierarchically organized with the interrelated topics contributing to one or few common themes. By removing distributional stopwords, we can reduce the overlapped connectivity between neighboring topics and thus improve the accuracy of boundary placement for topics.

However, as illustrated by [8], some distributional stopwords are not truly evenly distributed in that they can be reasonably focused in certain regions of the text. Referring to Figure 1, words like “form” are often identified as distributional stopwords, but if we divide the text into two regions and place the boundary at sentence 48, then within the first region, “form” behaves more like a content word, since its distribution is highly focused near the start of this region. Only within the second region does the word “form” act more like a distributional stopword. We call such words regional distributional stopwords in contract to the document-level distributional stopwords, since they are evenly distributed only within a region rather than over the entire text.

The existence of regional distributional stopwords suggests that when removing distributional stopwords at the document level, we should be more conservative, avoiding excessive removal of words that may carry useful content. In addition, we should apply distributional stopword removal recursively from the document level to a topic level so that we can capture regional distributional stopwords for the segmentation of subtopics. Note that when applying distributional stopword removal for a topic, we need to compute them from the original text, since a distributional stopword at the document level may be a useful content word at a topic level (e.g., "form" as a stopword at the document level vs. as a content word in the first region of a partition). Therefore, the combination of granularity for different segment sizes and distributional stopword removal at different segment levels should help improve the segmentation performance of coherent documents.

4 New Measures for Distributional Stopwords

Previous work by [4] and [5] focuses on the density and evenness of distributional stopwords. Given that word repetitions tend to be localized for a topic or a set of topics, we believe that the distribution coverage of a word is also important for selecting a distributional stopword: locally connected words should have a small coverage while evenly distributed stopwords should have a wider coverage. In this section, we propose two new measures based on the distribution coverage of words. We also modify an existing measure called distribution difference by relying on the density of words in a way similar to another measure called distribution significance [5].

4.1 Inverse Distribution Coverage

We use $|w_i|$ to denote the number of sentences that contain word w_i in a given document. Let us label these sentences with positions: $j_1, j_2, ..., j_{|w_i|}$. Then, the distribution coverage for word w_i can be defined as follows:

$$dc(w_i) = j_{|w_i|} - j_1 + 1 \tag{1}$$

In other words, the distribution coverage corresponds to the interval that spans over all occurrences of the word, including the first and the last sentences that contain the word.

Intuitively, the smaller the distribution coverage, the more specific the word for representing a topic. This approach represents the relevance of a word similar to a Zipfian distribution of word frequency in information retrieval, where a wide coverage indicates a highly distributed word that appears in many topics of a document. Thus, a word with a wide coverage resembles a stopword in information retrieval: it has low content and does not help discriminate between different documents. For this reason, we further define the inverse distribution coverage for a word as follows:

$$idc(w_i) = \frac{1}{dc(w_i)} \tag{2}$$

Based on this definition, a word spanning over a small interval will have a large inverse distribution coverage, thus carrying more content for a topic. On the other hand, a word covering a large interval will have a small inverse distribution coverage, which becomes a good candidate for distributional stopword removal.

4.2 Frequency-Adjusted Inverse Distribution Coverage

Although the distribution coverage captures the specificity of a word in representing a topic, it fails to distinguish the importance of a word at the same level of specificity. For example, a word that occurs more frequently should be more informative than another word with the same distribution coverage.

To combine the effects of both coverage and frequency of a word, we extend the inverse distribution coverage with the sentence frequency of a word, and call the new measure frequency-adjusted inverse distribution coverage:

$$fidc(w_i) = |w_i| \times idc(w_i) = \frac{|w_i|}{j_{|w_i|} - j_1 + 1} \tag{3}$$

The *fidc* measure resembles closely to the $TF \times IDF$ weight for document representation in information retrieval. The higher the sentence frequency $|w_i|$, the more important the word for a topic representation, and the smaller the inverse distribution coverage, the more specific the word. The two together help differentiate a good term (high content word) from a bad term (distributed stopword) for topic representation.

4.3 Distance-Based Distribution Difference

Vasak and Song proposed two measures for distributional stopword removal: distribution difference and distribution significance [5]. To combine the effects of these two measures, we propose the following distance-based distribution difference measure for distributional stopword removal. Given the sentence positions: $j_1, j_2, ..., j_{|w_i|}$, we add two dummy positions: $j_0 = 0$ and $j_{|w_i|+1} = m + 1$, where m is the total number of sentences in a given document. Then, the distance between occurrences k and $k + 1$ can be defined as: $dist(k, k + 1) = j_{k+1} - j_k$. The distance-based distribution difference is defined as follows:

$$dist_{dd}(w_i) = \left[\sum_{k=0}^{|w_i|} |dist(k, k+1) - \overline{dist}| \right] \times \frac{1}{|w_i|} \tag{4}$$

where $\overline{dist}$ is the average distance over all the individual distances for word w_i.

This combined measure is simpler than the original distribution difference since we do not need to identify the appropriate partition level and the corresponding block size. It is also more discriminative since we emphasizes the variance of distributed distances. Thus, the two together should help identify distributional stopwords. If a word is more or less uniformly distributed, the $dist_{dd}$ score will be low, making it a good candidate for distributional stopword removal.

5 Experiments

We conduct experiments to show the effectiveness of our new measures for distributional stopword selection and to determine the usefulness of regional distributional stopword removal for subtopic segmentation. The first experiment is focused on distributional stopword remvoal at the document level, while the second experiment is aimed at regional distributional stopword removal at a topic level. The results should help us understand better the effects of distributional stopword removal and bring us closer to the re-construction of the underlying hierarchical organization of a coherent document.

5.1 Data Sets

The Mars data set is made up of expository text that has a high vocabulary (the large number of unique words) for a specific domain. The data set consists of eight sections (Chapters 1 to 3) from the book "Mars" by Percival Lowell published in 1895 [4,5]. The sections differ in subject matter and segment granularity. On average, each section has 3 topic segments, 10 subtopic segments, 28 paragraphs, and 3,700 words. A typical topic segment has about 47 sentences and 1,235 words while a typical subtopic segment has about 14 sentences and 366 words.

Six annotators are involved in marking the Mars data set. The annotators are asked to mark paragraph boundaries at which the topics and subtopics are changed. The annotators' judgements are further reviewed since there are often disagreements among different annotators. A *true* boundary is set at a threshold of three or more judgments following Hearst's strategy [1]. The combined judgments are used to establish the basis for comparing different text segmentation algorithms.

To see the effects of distributional stopword removal at both document and regional levels, we extend the Mars data set by extracting topic segments that consist of subtopics and treating the topic segments as separate documents. The extended Mars data set has a total of 18 documents and on average each document has about 4 topic segments and 100 stemmed word types.

5.2 Evaluation

Our segmentation results are evaluated by the P_k measure proposed by Beefferman et al. [9]. The metric measures near misses by assigning a value to all boundaries by identifying whether a randomly chosen pair of words belong to the same segment or not. All word pairs are chosen to be exactly k words apart (half of the average length for reference segments). The probabilistic error metric of P_k is the proportion of the words which are wrongly predicted: either incorrectly placed into the segment or incorrectly placed outside the segment with regard to a reference segment.

The P_k measure scores zero if there is an exact prediction (perfect segmentation) between a hypothesized segmentation and the reference segmentation; otherwise, its value is in the range between 0 and 100. Since the P_k metric scores are not comparable against individual text due to different mean segment lengths, we use the average P_k for a set of testing documents to compare segmentation results.

5.3 Experimental Procedure

Each document is first tokenized and then pre-processed by sentence splitting [10], punctuation removal, general stopword removal (Choi's list [11]), and stemming (Porter's algorithm [12]).

Using any of the measures for distributional stopwords, we can rank all the stems in a document by their scores from the lowest to the highest. The higher the scores, the more content the stems; thus, we want to cut the stems with low scores and treat them as distributional stopwords. We use a rank-based value (such as lower 25%) to remove those stems whose ranks are below this threshold. We go through the following steps to find a suitable cutoff threshold:

1. For each document, rank all the stems according to the chosen measure for distributional stopword removal.
2. Divide the ranked stems into one hundred intervals: 0.00, 0.01, 0.02, $\cdots$, 0.99.
3. Cut the stems at each of the 100 intervals and compute the corresponding P_k values with the chosen method for text segmentation.
4. Choose up to 10 best intervals with lower P_k values that are better than the baseline result (which does not remove any distributional stopwords or use the rank value of 0.0).
5. Merge all the selected intervals for all the documents to form a pool of values for potential thresholds.
6. Test the pool of potential thresholds on all training documents and select the one with the lowest average P_k value. We denote the final cutoff threshold as RTh for the rank-based average threshold.

We repeat the above procedure for each of the following segmentation algorithms: Hearst's implementation of TextTiling based on similarity curve with default parameters [1], Choi's implementation of Reyner's $R98_{max}$ based on dotplot [2], Choi's implementation of C99 based on $R98_{max}$ with default parameters [11], and Utiyama and Isahara's U00 based on language models [3].

5.4 Experiment 1 – Document Distributional Stopword Removal

Our first experiment is focused on the document level distributional stopword removal. Segmentation results are obtained on the Mars data set. The main goal is to determine whether the coverage notion is better than the density and the evenness notions for distributional stopword removal. We generate results for existing and new measures on both topic and subtopic segmentations.

Results for Topic Segmentation. Table 1 shows the results of topic segmentation for each pair of distributional stopword measures and text segmentation methods. We show the average P_k values along with the corresponding rank-based threshold RTh values and the P_k gains with respect to the baseline results.

The segmentation results in Table 1 are measured against the topic marking in the reference segmentation. As can be seen, Text Tiling is not sensitive to distributional stopword removal, since there are no big differences between all the measures tested in

Table 1. Results of Topic Segmentation on Mars Data Set

Measures	*RTh*	*Tiling*	*RTh*	*C99*	*RTh*	*U00*	*RTh*	*R98max*
Baseline	62.56		56.82		57.00		33.48	
Ji and Zha	0.97	61.19	0.99	48.85	0.82	44.58	0.45	28.82
Difference	0.57	62.03	0.96	49.10	0.74	41.52	0.34	30.99
Significance	0.07	62.41	0.99	51.01	0.76	47.84	0.81	31.44
Distance Difference	0.24	62.31	0.91	35.56	0.01	57.0	0.09	32.93
Inverse Coverage	0.05	61.75	0.98	38.80	0.04	56.91	0.13	28.67
Freq Inverse Coverage	0.09	62.41	0.98	35.55	0.65	48.33	0.19	28.71

the experiments. This may be attributed to the default setting for the parameters in this program, which is set up for subtopic segmentation rather than topic segmentation.

However, all other segmentation methods have significant gains in performance when used with the distributional stopword measures. In particular, for C99, the three new measures perform much better than the old measures, with performance gains ranging 18.07–21.27% (compared with 5.81–7.97% for the old measures). For U00, the old measures remain better, probably due to the aggressive cuts for distributional stopwords with RTh thresholds ranging 74–82%. For both C99 and $R98_{max}$, the coverage-based measures (Inverse Distribution Coverage and Frequency-Adjusted Inverse Distribution Coverage) are more effective than the other density and evenness-based measures. The other new measure (Distance-Based Distribution Difference) only shows significant performance gain with C99, but not much improvements for the other segmentation methods.

Results for Subtopic Segmentation. Since some segmentation methods are suited for topic segmentation while others are geared for subtopic segmentation, we also conduct experiments on subtopic segmentation. The segmentation results in Table 2 are measured against the subtopic marking in the reference segmentation.

Table 2. Results of Subtopic Segmentation on Mars Data Set

Measures	*RTh*	*Tiling*	*RTh*	*C99*	*RTh*	*U00*	*RTh*	*R98max*
Baseline	51.98		48.32		43.94		47.86	
Ji and Zha	0.18	49.75	0.58	41.41	0.55	43.03	0.26	43.33
Difference	0.87	49.69	0.31	41.32	0.04	43.16	0.19	43.88
Significance	0.91	48.21	0.95	39.76	0.97	41.26	0.45	42.62
Distance Difference	0.90	49.23	0.82	40.02	0.38	39.93	0.49	46.12
Inverse Coverage	0.95	49.15	0.44	41.78	0.72	40.21	0.73	43.47
Freq Inverse Coverage	0.75	47.28	0.74	38.53	0.91	37.97	0.17	42.67

As can be seen in Table 2, Text Tiling indeed performs better for subtopic segmentation, since the baseline P_k value is reduced to 51.98 (compared with 62.56 for topic segmentation). In addition, all measures for distributional stopword selection are helpful, with the highest performance gain of 4.2% coming from Frequency-Adjusted Inverse Distribution Coverage.

Similarly, C99 and U00 also do better for subtopic segmentation, as indicated by the lower baseline P_k values. In terms of distributional stopword removal, all three new measures do well, with Frequency-Adjusted Inverse Distribution Coverage being the best performer (9.79% performance gain for C99 and 5.97% for U00). For C99, all measures for distributional stopword selection are effective (with the performance gains ranging 6.54–9.79%), while for U00, the three new measures are more effective than the three existing measures.

However, $R98_{max}$ does much worse for subtopic segmentation, indicating that this segmentation method is more suited for topic segmentation. The coverage-based measures (Inverse Distribution Coverage and Frequency-Adjusted Inverse Distribution Coverage) remain to be effective for this method, but the density and evenness-based measures are almost equally helpful (5.24% performance gain for Distribution Signification and 4.53% for Ji and Zha's measure).

5.5 Experiment 2 – Regional Distributional Stopword Removal

As introduced in Section 3, regional distributional stopwords are evenly distributed only within a region rather than over the entire text. To see their effects on text segmentation, we also conduct experiments on the extended Mars data set, which is obtained by extracting segments that contain subtopics and treating these topic segments as separate documents. The results in Table 3 are measured against the corresponding subtopic marking in the reference segmentation.

Table 3. Results of Text Segmentation on the Extended Mars Data Set

Measures	*RTh*	*Tiling*	*RTh*	*C99*	*RTh*	*U00*	*RTh*	*R98max*
Baseline	52.71		47.4		53.93		44.25	
Ji and Zha	0.98	44.98	0.98	41.75	0.70	44.58	0.27	41.68
Difference	0.52	47.27	0.98	41.52	0.92	45.94	0.12	41.75
Significance	0.94	46.51	0.99	42.19	0.76	46.45	0.61	40.96
Distance Difference	0.83	43.33	0.91	39.01	0.88	49.47	0.68	40.22
Inverse Coverage	0.66	44.59	0.97	38.98	0.52	50.40	0.36	41.21
Freq Inverse Coverage	0.98	44.82	0.99	39.52	0.63	45.58	0.43	42.08

Similar to the results in Table 2, Text Tiling performs well for subtopic segmentation, with a comparable baseline P_k value in Table 3. However, the impact of regional distributional stopword removal is stronger than that of document distrbutional stopword removal, since the performance gains are increased from 2.23–4.2% to 5.44–9.38% for all measures, with Distance-Based Distribution Difference being the best performer.

For C99, the new measures for distributional stopword selection perform better than the existing measures. In particular, the simple Inverse Distrbution Coverage does the best, indicating that the coverage of a term plays a more important role than the density or the evenness of a term as we move into the segmentation of subtopics.

For U00, we notice a similar pattern as in Table 1: the three existing measures and Frequency-Adjusted Inverse Distribution Coverage are more effective than the other two new measures for distributional stopword selection. In addition, the impact of regional distributional stopword removal is stronger than that of document distributional stopword removal when compared with the results in Table 2.

Finally, for $R98_{max}$, all measures are helpful, but the differences among them are not big. Overall, $R98_{max}$ tends to have lower P_k values across all the experiments, and as a result, the effect of distributional stopword removal is not as strong as that for other text segmentation methods.

6 Conclusions and Future Work

Our experimental results show that distributional stopword removal is generally useful and desirable for text segmentation of coherent documents. In terms of distributional stopword selection, the two new coverage-based measures (Inverse Distribution Coverage and Frequency-Adjusted Inverse Distribution Coverage) are not only efficient to compute, but aslo more accurate than the other density and evenness-based measures. In particular, Frequency-Adjusted Inverse Distribution Coverage is often the best performer, since it emphasizes both the specificity and the importance of a word. The other new measure Distance-Based Distribution Difference, although conceptually general, is mostly effective for C99 algorithm. In terms of text segmentation methods, C99 and U00 benefit more from distributional stopword removal, and especially for C99, the performance gain is as high as 21.27%. In terms of subtopic segmentation, regional distributional stopword removal is also effective, and for Text Tiling and U00, it has stronger effects than document distributional stopword removal.

For future work, we plan to apply our measures for distributional stopword removal to a wider spectrum of data sets (such as transcribed lecture notes and scientific articles) with a larger number of documents. We are also interested in recursively applying our measures for distributional stopword removal to re-build the hierarchical organizations of coherent documents. Finally, we want to examine the interactions between the local and distributed words from the view point of emphasis (as is seen in figure 1 where words like “life” and “planet” can potentially be used to distinguish betweeen different segments) to generalize a measure for distributional stopword removal.

Acknowledgements

We acknowledge the financial support of CITO, a division of Ontario Centres of Excellence (OCE). We would also like to thank the annotators for marking the Mars data set, Masao Utiyama and Marti Hearst for their text segmentation systems, and Freddy Choi for his publicly available toolkit.

References

1. Hearst, M.: Multi-Paragraph Segmentation of Expository Text. In: Proceedings of the ACL, pp. 9–16 (1994)
2. Reynar, J.C.: Topic Segmentation: Algorithms and Application. Ph.D. Thesis, University of Pennsylvania (1998)

3. Utiyama, M., Isahara, H.: A Statistical Model for Domain-Independent Text Segmentation. In: Proceeedings of the ACL, pp. 491–498 (2001)
4. Ji, X., Zha, H.: Domain-Independent Text Segmentation Using Anisotropic Diffusion and Dynamic Programming. In: Proceedings of the ACL, pp. 322–329 (2003)
5. Vasak, J., Song, F.: Word Distribution Based Methods for Minimizing Segment Overlaps. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 491–498. Springer, Heidelberg (2007)
6. Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, New York (1976)
7. Skorochod'ko, E.F.: Adaptive Method of Automatic Abstracting and Indexing. In: Proceedings of the IFIP, vol. (71), pp. 1179–1182 (1972)
8. Malioutov, I., Barzilay, R.: Minimum Cut Model for Spoken Lecture Segmentation. In: Proceedings of the ACM SIGIR, pp. 25–32 (2006)
9. Beeferman, D., Berger, A., Lafferty, J.D.: Statistical Models for Text Segmentation. Machine Learning 34(1-3), 177–210 (1999)
10. Reynar, J., Ratnaparkhi, A.: A Maximum Entropy Approach to Identifying Sentence Boundaries. In: Proceedings of the ANLP, pp. 16–19 (1997)
11. Choi, F.Y.Y.: Advances in Domain Independent Linear Text Segmentation. In: Proceedings of the NAACL, pp. 26–33 (2000)
12. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)

Using TectoMT as a Preprocessing Tool for Phrase-Based Statistical Machine Translation

Daniel Zeman

Univerzita Karlova v Praze, ÚFAL,
Malostranské náměstí 25, 11800 Praha, Czechia
zeman@ufal.mff.cuni.cz
http://ufal.mff.cuni.cz/~zeman/

Abstract. We present a systematic comparison of preprocessing techniques for two language pairs: English-Czech and English-Hindi. The two target languages, although both belonging to the Indo-European language family, show significant differences in morphology, syntax and word order. We describe how TectoMT, a successful framework for analysis and generation of language, can be used as preprocessor for a phrase-based MT system. We compare the two language pairs and the optimal sets of source-language transformations applied to them. The following transformations are examples of possible preprocessing steps: lemmatization; retokenization, compound splitting; removing/adding words lacking counterparts in the other language; phrase reordering to resemble the target word order; marking syntactic functions. TectoMT, as well as all other tools and data sets we use, are freely available on the Web.

Keywords: phrase-based translation, preprocessing, reordering

1 Introduction

It is widely accepted that linguistically informed preprocessing of training data can improve quality of statistical machine translation. The general goal is, in most cases, to make the source and the target texts grammatically more similar and thus easier to learn for a statistical machine translation system. Both source and target languages can be preprocessed. The task is easier if we restrict preprocessing to the source language. In this case, the source part of the training parallel data is preprocessed in the hope that the resulting string can be better aligned with the target string and thus better phrase translation model can be learned. During the decoding phase (i.e. applying the model to new unseen data), the source test corpus is preprocessed exactly the same way, then the model is applied.

If we choose to preprocess the target side of the training data, we need to be able to reverse the transformation in a postprocessing step after the decoding phase. The assumption is that the model trained on the preprocessed data will produce output similar to the preprocessed data. However, the required output of the MT system is natural, unpreprocessed target language string. On the

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 216–223, 2010.

other hand, we cannot rely on any expected structure of the output, as the MT system can (and will) make errors.

There are several reasons for considering preprocessing of parallel corpora:

- Richer morphology on one side results in sparse data. For instance, the English word *woman* may appear in singular or in plural *(women)*. In contrast, its Czech translation *žena* is marked for number and case, resulting in 10 distinct forms *(žena, ženy, ženě, ženu, ženo, ženou, žen, ženám, ženách, ženami)*. It is not realistic to expect that each of these forms will occur frequently enough in the training data, with every possible English translation. By separating morphology from the lexical information the data sparseness can be reduced. Similar effect can be achieved by separating compound words (e.g. in German) or separating morphemes that would be standalone words in the target language (e.g. in Arabic-to-English translation).
- If the target language is the morphologically richer of the two, generating source pseudowords bearing necessary information such as syntactic functions (subject, object etc.) can help to figure out the correct target word forms. For other target languages, pseudowords can help generate target words that normally do not have direct source counterparts. For instance, pro-drop languages such as Czech or Spanish do not require that personal pronouns are present when otherwise there would be no subject. However, verbs in such languages are often marked for person, which can help with generating the correct personal pronoun in the non-pro-drop target language. Thus, in Czech-to-English MT, we have to learn *jdu* → *I go*, *jdeš* → *you go* etc. The English personal pronouns can be more easily generated if we augment the Czech source with explicit person+number information, supplied by morphological analysis of the Czech verbs.
- Significant differences in word order between the two languages. Preprocessing includes syntactic parsing of the source language, then the phrases are reordered according to some rules. The availability of a parser is crucial here because whole phrases have to be moved along, not just words.

The following transformations are examples of possible preprocessing steps: lemmatization; retokenization, compound splitting; removing/adding function words that systematically lack counterparts in the other language (articles, personal pronouns etc.); reordering of phrases in parsed source sentence in order to make the word order closer to that of the target language; adding pseudo-tokens for syntactic functions such as subject, predicate, object.

There have been numerous publications on various aspects of preprocessing for several language pairs. In this paper, we present a systematic comparison of preprocessing techniques for two language pairs: English-Czech and English-Hindi. Due to the reasons mentioned above, we restrict ourselves to preprocessing of the source language. The two target languages, although both belonging to the Indo-European language family, show significant differences in morphology, syntax and word order. We describe how TectoMT, a successful framework developed originally for deep-syntax-based machine translation, can be used as preprocessor for a phrase-based MT system, such as Moses or Joshua. We compare the two language pairs and the optimal sets of source-language transformations applied to them.

The rest of the paper is organized as follows: in Section 2 we summarize the related work, in Section 3 we introduce TectoMT and other software, in Section 4 we describe the transformations used for each language. Then we describe the data sets (Section 5) and discuss preliminary results (Section 6).

2 Related Work

There exists a body of previous work that is related to ours in one or more aspects. We discuss a selection of related publications in this section.

Nießen and Ney [1] describe a German-to-English MT system that integrates morphology-based preprocessing of German. They split German compound words, join separable verb prefixes with verbs and augment German words with morphological information. They observe that while many German morphological features (such as the distinction between the nominative and the accusative) are not reflected in English, sometimes more morphological information is present in the English word than in its German counterpart: *das Zimmer* → *the room* vs. *die Zimmer* → *the rooms.*

Collins, Koehn and Kučerová [2] also experiment with German-to-English SMT. They use a syntactic parser to obtain an analysis of the source language string, then they apply a series of transformations to the parse tree, effectively reordering the source string. The goal of this step is to recover an underlying word order that is closer to the target language word order than the original string. They report a statistically significant improvement of the BLEU score on the Europarl corpus.

Popović et al. [3] present results on a very small Serbian-English corpus for both translation directions, sr-en and en-sr. For each direction, they preprocess the source side of the corpus. English preprocessing is limited to the removal of articles. Serbian preprocessing consists of two steps: lemmatization and special treatment of verbs (person verb feature is used to generate missing personal pronoun).

Goldwater and McClosky [4] discuss the Czech-to-English task on the Prague Czech-English Dependency Treebank. To reduce the data sparseness problem, they first lemmatize the source Czech text, than attempt to partially restore the lost information by introducing pseudowords or separated morphemes.

Different issues are encountered in Arabic-to-English translation (Habash and Sadat [5], El Isbihani et al. [6]). Here the preprocessing mostly involves English-like retokenization of Arabic (comparable to the compound splitting in German), i.e. separating conjunctions, prepositions and articles that are normally written jointly with the noun.

Prokopová [7] investigates various ways of enriching Czech input in Czech-to-English translation. Besides word reordering (to get the fixed English subject-verb-object word order), she also inserts into the Czech string frequent English words that may not have any counterpart in Czech: articles, personal pronouns, the infinitival marker *to*, prepositions *of* and *by*.

Avramidis and Koehn [8] use parse trees of the source English text not to reorder it but rather to acquire information about syntactic functions of the English words. That information can then be made explicit and help generate the correct case marker in the target language within an English-to-Greek MT experiment. Reduction of errors in verb conjugation and in noun case agreement is reported.

Axelrod et al. [9] present another experiment with German stemming and compound splitting but this time for a German-to-Spanish MT system.

Popović et al. [10] apply part-of-speech-based (i.e., no parsing) reorderings of the source language to the German-, French- and Spanish-to-English tasks. Again, German compound splitting is found helpful, too.

Finally, Ramanathan et al. [11] address the large word-order discrepancy in English-to-Hindi MT, along with richer morphology of the target language. They use preprocessing to figure out the English syntactic functions and to get the target SOV word order; they also use postprocessing to generate Hindi case markers and suffixes.

In general, former work focused more on translation to English (which usually meant into the morphologically poorer language) than on translation into a morphologically rich language; however, the interest in the latter has been increasing recently.

3 TectoMT and Related Tools

TectoMT [12] is a highly modular NLP framework implemented in Perl under Linux. It was originally developed to facilitate machine translation within the classical analysis-transfer-synthesis paradigm. It is composed of numerous reusable processing modules (called "blocks"), which are equipped with uniform object-oriented interfaces. Some of the blocks wrap large NLP applications such as taggers and parsers (together with pre-trained models), others are designed to perform tiny specialized operations: for instance, operating on output of a particular parser, a block can apply some heuristics to correct treatment of coordination. Unified application programming interface allows for rapid development of such language transformations without having to care about the file format, task parallelization etc. Because of the unique modular environment, the usefulness of TectoMT extends beyond machine translation to virtually any natural language processing task.

We use TectoMT to analyze the English side of the parallel corpora. We do not use the transfer- and generation blocks of TectoMT; instead, we train a phrase-based SMT system on the preprocessed corpora. Two blocks wrapped in TectoMT deserve being mentioned separately: the morphosyntactic tagger Morče [13] and the MST (maximum spanning tree) dependency parser[14]. Besides and around these two, we reuse nearly 40 other blocks that the TectoMT developers designed and routinely use to improve the analysis of English texts. On top of it, our reordering block takes care for the transformations described below.

As the phrase-based SMT component, we use Joshua [15].

4 Overview of Transformations

4.1 English to Czech

Articles. There are no definite or indefinite articles in Czech. The SMT systems waste energy to align them to Czech, sometimes it makes the data unnecessarily sparse: e.g., Czech *pražskou* has two English counterparts, *the Prague* and *Prague the.* Solution: All occurrences of the words *a, an, the* tagged `DT` are removed.

Target case selection. English almost completely lacks the notion of grammatical case (except for the direct and oblique cases of pronouns). In Czech, there are 7 cases. In general, it is not easy to select the correct case (see also *Target agreement* below), however, the subject is typically in nominative. Hence appending `/Sb` to the root word of the English subject (provided we have parsed the English input) can help to generate the nominative on the Czech side.

Target agreement. It is difficult to generate target phrases that agree in gender, number and case as required by Czech grammar. For instance, English *trading day* can be translated as nominative *obchodní den*, genitive *obchodního dne*, dative *obchodním dni* etc. If the SMT system does not learn the Czech phrase in all the cases, it will attempt to translate each word separately, in which case however it will lose the agreement feature. Thus, incorrect translations such as **obchodním dne* are frequently seen in the output. The solution is more tricky in this case. We could separate lemma from the morphological features in the Czech text; however, this would mean preprocessing of the target text, which we prefer to avoid. We leave this problem open for further research.

Verbal groups. English has many analytical tenses of verbs and is richer than Czech in that respect. To make it easier for phrase-based SMT systems to get the correct tense, we move all auxiliaries, modal verbs etc. as close to the main verb as possible. Example: *will only make matters worse* → *only will make worse matters.*

Personal pronouns. English personal pronouns functioning as subjects are joined with their verbs. Word alignment tends to align them to Czech verbs anyway; however, there is room for mis-alignments and data sparseness is unnecessarily increased.

4.2 English to Hindi

Articles. Similarly to Czech, there are no definite articles in Hindi. However, indefinite articles are sometimes translated using the numeral एक *(eka)* (“one”). Solution: Remove occurrences of *the* from the English text.

Postpositions. English prepositions are usually translated as postpositions in Hindi. Sometimes the postpositions are compound and they require concrete case ending for the preceding noun or pronoun. Examples: *in the house* → घर में *(ghara meṁ)* ("house in"), *my teacher's book* → मेरे अध्यापक की किताब *(mere adhyāpaka kī kitāba)* ("my-oblique teacher of-fem book"), *towards Ram* → राम की तरफ़ *(rāma kī tarafa)* ("Ram of-fem direction"). Solution: Convert English prepositions to postpositions, i.e. move them after the noun phrase they govern.

Subject-object-verb order. English is an SVO language while Hindi is an SOV language, i.e. Hindi verbs occur mostly at the end of the clause, as in:

I'm doing some work with a friend.
एक मित्र के साथ कुछ काम कर रहा हूँ ।
(eka mitra ke sātha kucha kāma kara rahā hūṁ .)
"one friend of-masc with some work do -ing-masc I-am ."

Solution: reorder clauses so that the main finite verb goes to the end.

To have. There is no direct equivalent in Hindi for the English verb *to have.* Instead, various indirect constructions are used to convey the sense of having. Example:

We have time.
हमारे पास समय है ।
(hamāre pāsa samaya hai.)
"our-oblique at time is."

Solution: Make *to have* an exception to the verb reordering rule introduced above. Keep it with its subject and let the SMT learn translations like *we have* → हमारे पास *(hamāre pāsa)*, *X has* → X के पास *(X ke pāsa)* etc.

5 Data

All test sets mentioned in this section have only one reference translation per sentence.

5.1 English to Czech

We use the News Commentary 10 corpus (94,697 sentence pairs) for training, the WMT 2008 test set (2,051 sentence pairs) for development and the WMT 2009 test set (2,525 sentence pairs) for testing. All these corpora are freely available at `http://www.statmt.org/wmt10/translation-task.html`.

5.2 English to Hindi

We use a cleaned version of the IIIT Tides corpus. This dataset was originally collected for the DARPA-TIDES surprise-language contest in 2002, later refined at IIIT Hyderabad and provided for the NLP Tools Contest at ICON 2008 [16]. The corpus is a general domain dataset with news articles forming the greatest proportion. It is aligned on sentence level, and tokenized to some extent. There are 50K sentence pairs for training, 1K pairs for development and 1K for testing.

Table 1. Translation from preprocessed English to Czech/Hindi.

Method	BLEU
English-Czech, Baseline	0.0863
English-Czech, Preprocessed	0.0905
English-Hindi, Baseline	0.1006
English-Hindi, Preprocessed	0.1029

6 Results

We evaluate the impact of the preprocessing transformations in two ways. A manual evaluation focuses at the phenomena described in Section 4. Human inspection of 50 sentence pairs selected randomly from the test data revealed that the case selection in Czech, and the alignment in both Czech and Hindi (with the reordered English) improved.

The newly aligned corpora were then used to train new translation models for the Joshua decoder and BLEU score has been used to evaluate the accuracy of the new models on test data. Unfortunately, all quantitative improvements so far are statistically insignificant. Table 1 presents the BLEU scores; as the impact of the transformations is rather low, we do not present detailed figures for isolated transformations.

Space limitations of this paper do not allow to describe all details of error analysis; however, translations generated by the model for the test data seem to suggest that morphology generation on the target side is a more important source of errors than the alignment problems addressed by our transformations.

7 Conclusion

We proposed a number of source-side transformations of English text in order to make it grammatically more similar to the target language, namely Czech and Hindi. We gave a comprehensive overview of related work and argued that TectoMT, a modular NLP framework, is a tool highly suitable for the preprocessing task. Different sets of transformations were proposed w.r.t. the given target language. The preprocessed corpora led to better word alignment but not to significant improvement of translation quality in terms of BLEU score. Future research will focus on morphology of the target language, which seems to be a more important source of errors.

Acknowledgments. The work on this project has been supported by the grant MSM0021620838 by the Czech Ministry of Education.

References

1. Nießen, S., Ney, H.: Statistical Machine Translation with Scarce Resources Using Morpho-Syntactic Information. Computational Linguistics 30(2), 181–204 (2004)

2. Collins, M., Koehn, P., Kučerová, I.: Clause Restructuring for Statistical Machine Translation. In: Proceedings of the 43rd Annual Meeting of the ACL, pp. 531–540. ACL, Ann Arbor (2005)
3. Popović, M., Vilar, D., Ney, H., Jovičić, S., Šarić, Z.: Augmenting a Small Parallel Text with Morpho-Syntactic Language. In: Proceedings of the ACL Workshop on Building and Using Parallel Texts, pp. 41–48. ACL, Ann Arbor (2005)
4. Goldwater, S., McClosky, D.: Improving Statistical MT through Morphological Analysis. In: Proceedings of HLT-EMNLP, pp. 676–683. ACL, Vancouver (2005)
5. Habash, N., Sadat, F.: Arabic Preprocessing Schemes for Statistical Machine Translation. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pp. 49–52. ACL, New York (2006)
6. El Isbihani, A., Khadivi, S., Bender, O., Ney, H.: Morpho-syntactic Arabic Preprocessing for Arabic-to-English Statistical Machine Translation. In: Proceedings of the Workshop on Statistical Machine Translation, pp. 15–22. ACL, New York (2006)
7. Prokopová, M.: Automatic Simplification of Texts for Translation. Master's thesis, Univerzita Karlova v Praze, Praha, Czechia (2007)
8. Avramidis, E., Koehn, P.: Enriching Morphologically Poor Languages for Statistical Machine Translation. In: Proceedings of ACL 2008: HLT, pp. 763–770. ACL, Columbus (2008)
9. Axelrod, A., Yang, M., Duh, K., Kirchhoff, K.: The University of Washington Machine Translation System for ACL WMT 2008. In: Proceedings of the Third Workshop on Statistical Machine Translation, pp. 123–126. ACL, Columbus, Ohio (2008)
10. Popović, M., Vilar, D., Stein, D., Matusov, E., Ney, H.: The RWTH Machine Translation System for WMT 2009. In: Proceedings of the Fourth Workshop on Statistical Machine Translation, pp. 66–69. ACL, Athîna, Greece (2009)
11. Ramanathan, A., Choudhary, H., Ghosh, A., Bhattacharyya, P.: Case Markers and Morphology: Addressing the Crux of the Fluency Problem in English-Hindi SMT. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 800–808. ACL and AFNLP, Suntec (2009)
12. Žabokrtský, Z., Ptáček, J., Pajas, P.: TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In: Proceedings of the Third Workshop on Statistical Machine Translation, pp. 167–170. ACL, Columbus (2008)
13. Votrubec, J.: Selecting an Optimal Set of Features for the Morphological Tagging of Czech. Master thesis, Univerzita Karlova v Praze, Praha, Czechia (2005)
14. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective Dependency Parsing using Spanning Tree Algorithms. In: Proceedings of the Human Language Technology / Empirical Methods in Natural Language Processing conference (HLT-EMNLP), pp. 523–530. ACL, Vancouver
15. Li, Z., Callison-Burch, C., Khudanpur, S., Thornton, W.: Decoding in Joshua: Open Source, Parsing-Based Machine Translation. The Prague Bulletin of Mathematical Linguistics 91, 47–56
16. Venkatapathy, S.: NLP Tools Contest – 2008: Summary. In: Proceedings of ICON 2008 NLP Tools Contest, Pune, India (2008)

Automatic Sentiment Analysis Using the Textual Pattern Content Similarity in Natural Language

Jan Žižka and František Dařena

Department of Informatics – SoNet Research Center
Faculty of Business and Economics, Mendel University in Brno
Zemědělská 1, 613 00 Brno, Czech Republic
{zizka,darena}@mendelu.cz

Abstract. The paper investigates a problem connected with automatic analysis of sentiment (opinion) in textual natural-language documents. The initial situation works on the assumption that a user has many documents centered around a certain topic with different opinions of it. The user wants to pick out only relevant documents that represent a certain sentiment – for example, only positive reviews of a certain subject. Having not too many typical patterns of the desired document type, the user needs a tool that can collect documents which are similar to the patterns. The suggested procedure is based on computing the similarity degree between patterns and unlabeled documents, which are then ranked according to their similarity to the patterns. The similarity is calculated as a distance between patterns and unlabeled items. The results are shown for publicly accessible downloaded real-world data in two languages, English and Czech.

Keywords: sentiment/opinion analysis, textual patterns, natural language, textual document similarity, similarity ranking.

1 Introduction

In this article, the authors present a procedure how to simply single out textual documents representing a specific opinion from a big group of documents with different opinions. The whole document group is centered around one common topic. This problem in question is now very actual. The last years have introduced various new web-technologies as Web 2.0, which facilitates a massive expansion of using the Internet for expressing different or miscellaneous opinions (via messages) between people, groups of people, administrative authorities, commercial or non-commercial organizations and institutions, and so like. Shared on-line journals known as *blogs* enable people to post daily entries about their personal experiences and hobbies. Customers can provide valuable feedbacks both for e-shops (like *amazon.com*) and their potential future customers who can read opinions and experiences willingly provided by shoppers. Similarly, the Internet is a host of uncountable discussion groups, newsgroups, and tens of social networks like *Facebook* that unites more than 100 millions of unique visitors. The significant part of the reciprocal communication data is expressed in natural languages using the textual form. Such the data are very interesting because they usually hide a lot of information and knowledge that can be used for various goals,

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 224–231, 2010.

including the business ones. Therefore, the question is how to mine that textual data. During the last 20 years or so, many useful text-mining methods and algorithms have been developed [1]. If the data enable the application of classification, it is the best approach. However, very often the potential training data miss labeling that is necessary for the future including of individual items (text documents) into their appropriate classes [2].

2 Basic Ideas

The article points to an alternative method that avoids the large labeling of individual training items. This method, described in the next chapters, introduces more initial information into the process of recognizing unlabeled items. The idea comes from a typical real situation when somebody has only a small collection of 'good' examples, or *patterns*, available. From a large number of various text documents, he or she wants to remove what is different from the patterns so that the remaining part of textual items represents interesting or relevant group of articles, blog entries, or discussion submissions. The main idea is based on determining the *similarity* degree [3] between the patterns and an unlabeled textual item [4]. In addition, this approach is not aimed at categorizing the items according some given topics. Typically, there is only one main shared topic and the authors are interested in separating the unlabeled text items according different *opinions* (or, as it is often used as an alternative term, *sentiments*). The automatic sentiment/opinion analysis can sort those numerous individual contributions that are expressed in natural languages. For example, people purchasing a specific product using an on-line web-based shop can later write their opinions [5]. Other purchasers can then submit similar or different meanings, also with a reference to previous entries, and the e-seller can analyze such submissions and draw appropriate conclusions. Similarly, in various discussion groups, some people can provide a message about their opinion of a book, film, politician, and so like – it can help others for their final decisions if yes or no. Thus, a reader of such text messages may want to select only positive or negative items from a large collection. He or she can choose some patterns which correspond to his or her sentiment and then let a machine find similar items.

3 Data Sets Used for Experiments

The main idea of looking for documents that are similar to patterns was tested using several real-world data sets obtained by downloading from publicly and commonly accessible data sources: *amazon.com* [6] plus *amazon.co.uk* [7] (in English), and the archive of a Czech newspaper *MF Dnes* articles (the electronic version) [8] plus a website of a Czech political party [9] (both in Czech). The selected *amazon* data included opinions of customers on a particular iPod head-phones (consumer electronics) – 30 positive and 48 negative customer-opinions; computer hardware (Western Digital USB external hard disk) – 47 positive and 37 negative opinions; a book (King James version of Bible) – 130 positive and 86 negative opinions; a film (Caligula, the Unrated Edition)

– 100 positive and 64 negative opinions; and articles on a political party program (the Czech Social Democratic Party, ČSSD) – 22 positive and 22 negative opinions.

The *amazon* data-sets consist of various entries describing consumers' meanings. In addition, each contributor assigns one star (the worst evaluation of a product) up to five stars (the best evaluation) according to his or her opinion. The authors selected only two groups: with one- (negative) and five-star (positive) evaluation, excluding the rest because 2, 3, and 4 stars expressed more-or-less ambiguous opinions – it was also difficult for humans to decide whether it was a positive or negative opinion as those opinions were more-or-less mixed. The goal was an attempt to automatically select contributions belonging to one- or five-star evaluation using only the text contents.

It is necessary to remark that the *amazon* data are typical, with many expected problems. First, some contributions are long enough to clearly express the opinion, but most of them are short, some of them very short. Second, some positive (or negative) contributions express the same opinion using different words, for example, synonyms, idioms, or word connections – it means that such documents can look different from the word-based viewpoint. Third, the contributions contain mistypings and spelling errors, therefore the same word typed differently by two persons looks absolutely different from a machine point of view, extending the dictionary. To name but a few from, for example, the Western Digital USB hard-disk data: *absoloutely/absolutley/absolutely*, *beleive/believe*, *definately/definitely,* where only the last word form is correct. Anyway, these problems are quite typical (and naturally inevitable) for such blogs and discussion groups. The decision to experiment also with Czech texts was intended to show whether the presented method works also for other language than English. As an example, the authors found politically-based data that contained contributions *for* and *against* one of the present-day Czech political party platforms before elections. The realization and results of experiments with all the above mentioned data sets are described in the following sections.

4 Text Document Pre-processing and Representation

Textual documents in a natural language contain words and these words are used for determining the similarity between two documents. The preceding research of many scientists brought a lot of algorithms and methods that proved their good quality and usefulness also in practice. A standard procedure includes pre-processing based on creating a *bag-of-words* and consecutively a *dictionary* (a set of words) from words in text documents available. A certain advantage is that such a process can be easily done by computers. On the other hand, a bag-of-words contains words without the information about their original positions in a document – a loss of information. More sophisticated methods need more complex procedures that are – unlike the simple bag-of-words approach – much more dependent on a specific language. The authors decided to employ the standard procedure because they wanted to create a method based on the automatic natural-language processing without a big dependency on individual languages.

The pre-processing did not use removing stop-words because most of the items were very short and it would be necessary to carry out the specific stop-word analysis for each of the data collection – not always the list of common stop-words can be used. The shortness of text items was also the reason why the pre-processing procedure did not remove words with too high or too low frequency. The authors do not exclude the possibility that a deeper analysis of the text properties could somehow improve the results but it was not the primary goal of this paper.

Analogously, the ‘short’ words (typically, up to three letters) were not removed as it is typical for English texts because the authors wanted to apply the same approach to another (Slavonic) language, as well. Such a procedure would request additional text analysis that could be specific for different languages.

A word can be represented by several popular methods [1]: as a binary number (1/0, or a word is/is not in a document), frequency number (how many times a word is in a document), or $TF \times IDF$ (term frequency times the inverted document frequency). Independently on the representation, the dictionary (as well as each document) was transformed into a multidimensional vector where individual word representations were used as coordinates within the abstract space with each dictionary word as one of dimensions. Several initial experiments showed that the best results were provided by the frequency representation.

It is necessary to emphasize that the vectors did not contain the original class membership (that is, belonging to the positive or negative group of opinions).

5 The Similarity of Unlabeled Documents to Patterns

The similarity degree between an unlabeled document and a selected pattern is computed using the word representation. Here, the word frequency representation was used as features in the vectors, where a vector represents one text document. The more words with their frequencies are equal in a pair of vectors the higher the similarity between both documents is, and vice versa. Each document can be taken as a point in an n-dimensional abstract space with coordinates given by the word frequencies (each word makes one of the axes). In this case, the similarity degree can be expressed as a distance between two points where the zero distance means an identical couple of documents [1]. Therefore, having two textual documents A and B, the Euclidean distance L_E between a text document $d^{(A)}$ and $d^{(B)}$ in Cartesian coordinates can be calculated in the following way:

$$L_E = \sqrt{\sum_{i=1}^{n} (d_i^{(A)} - d_i^{(B)})^2}\ , \tag{1}$$

where $d_i^{(.)}$ is the i-th word coordinate of an n-dimensional point representing a document, and n is the number of unique words in the dictionary of all documents used as examples. The values of $d_i^{(.)}$ are the mentioned word frequencies.

If a document is taken as a vector in Cartesian coordinate system, then the vector similarity is given by the angle α between both vectors, where the zero angle means the 100% similarity. Typically, the vectors are very sparse, containing only few shared words between documents A and B. The similarity is expressed using *cosine* of the angle α:

$$cos(\alpha) = \frac{A \cdot B}{\parallel A \parallel \parallel B \parallel}, \tag{2}$$

which is a dot-product of two vectors A and B using the word frequencies as the features, where $\parallel . \parallel$ is a vector magnitude (length).

The other question is how many patterns are necessary as the good samples of an opinion which should be extracted from the whole data collection. It depends on the available number of examples that express the monitored kind of opinion. Too low example number may restrict the extracted documents too much. Too many different examples of a certain opinion may be the leading cause of too broad extraction with too many documents having a low similarity. Or, if there are too many very similar patterns, it can lead to a high and redundant computational complexity coming out from computation of many distances.

6 Ranking by Similarity

After computing distances of all unlabeled text documents to the patterns, the documents are sorted so that the most similar one is at the top of the rank and the less similar ones are placed lower. Then, the user may decide how many documents from the rank top could be processed by any following method. Typically, a user chooses a relatively small number, units or tens, of the top-ranked items for his or her following work. The rank can contain thousands or far more documents but the user can process only a limited number of them – naturally, the user wants the items that are very similar to the known patterns. Such a procedure is similar to 'classification' when only one class is known while all the available unlabeled items can belong to the unknown number of various classes.

As it can be expected, the suggested procedure cannot be errorless. The error of the similarity ranking originates in the fact that at the top of the rank there can also be false positives because the similarity is measured imperfectly and the noisy text items may sometimes cause improper results. Testing the procedure with known data, it is possible to study how many correct items are at the top and how the correctness decreases towards the bottom. Ideally, all items with the monitored opinion should be higher than items with different opinions. However, as the similarity decreases, the possibility of errors increases – in fact, also human beings often cannot quite unambiguously decide if an item still belongs among the relevant ones or not.

7 Experiments and Their Results

In the experiments with the five data sets mentioned above, randomly selected subsets of textual items with a specific opinion were used as patterns. The number of patterns changed from 10 to 35 (for details, see also the graph in Figure 1 and Table 1 description), depending also on the size of the original data set to keep the sufficient number of testing examples. For the *amazon* data, the authors focused on the radical univocal opinion degrees: to reveal whether it would be simply possible to correctly assign positive opinions written in a natural language to the five-star (or one-star) category. After detaching the patterns, the rest of the opinion examples was used for testing – the set of unlabeled items containing both the same monitored opinion, and the unmonitored rest (the opposite opinion).

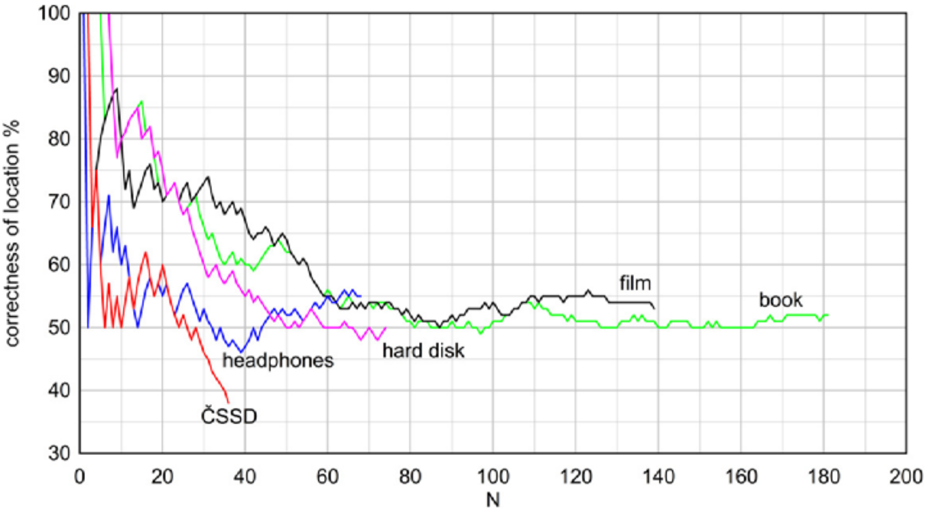


Fig. 1. With the increasing number N of documents taken from the top of the rank, the chance of obtaining misplaced items increases as well. At the beginning, for $N < 10$, some curves overlap but all start at $N = 1$. The ČSSD curve is for the political discussion data in Czech. The remaining four curves demonstrate positive and negative opinions in English written by *amazon* customers on various purchased products.

Applying the similarity procedure described above, the degree of errors was studied in dependence on the number of patterns and the type of the similarity as the parameters. In the graph Fig. 1, a reader can see the results for the optimal parameters. The experiments used the two similarity methods (*Euclidean* and *cosine*) with a small negligible difference between them. In some cases the *Euclidean* distance provided slightly better results than the *cosine* one. Thus, only the results with the *Euclidean* similarity are demonstrated here.

Each member of the testing set was successively compared to all the individual patterns. As the final similarity degree, the nearest pattern was taken, where the degree was given by the computed distance. After processing of all the testing samples, they were ranked according their similarity degree starting with the most similar ones at the top. Then, the number of misplaced items specified the error. Ideally, all the samples with the monitored opinion should be placed before the first item having the opposite opinion. However, some items were misplaced, especially those ones that were too brief, having not many words. The experiments showed that larger texts provided lower errors, which could be logically expected because of the higher information contents given by words. The curves in the graph Fig. 1 depict the percentage of correct opinion-placing (the vertical coordinate axis *correctness of location*) for the first N documents in

Table 1. Parameters of the individual data groups. The table shows the total number of samples, the number of samples with the positive (+) and negative (−) opinion, and the number of patterns with either the positive or negative monitored opinion. The negative monitored opinion took patterns from the negative samples and vice versa.

data group	*samples*	+ *opinions*	− *opinions*	*patterns*	*monitored opinion*
headphones	78	30	48	10	−
hard disk	84	47	37	10	+
book	216	130	86	35	+
film	174	110	64	25	+
ČSSD	44	22	22	8	+

the rank (N, the horizontal axis). For example, if a user would select the first five items ($N = 5$) from the top, there may be only examples of his or her monitored opinion type: the correctness would be 100 %. Analogously, for the first $N = 10$ items with three misplaced ones, the correctness is 70 %. As the N increases, the correctness expectedly decreases because the similarity of all samples varies and sometimes overlaps for different opinions.

The table Tab. 1 shows the detailed parameters for each of the five individual data groups. It is obvious that the data sets having more examples for both of the investigated opinion groups (positive and negative) provided better results: see the curves *book* and *film*. Such the data could give also more patterns for the monitored opinion (35 and 25, resp.). Clearly, their correctness descent is slower than for the other, smaller data. The political data gave the best results for 8 patterns, which is obviously not too many comparing with the other data groups, and it would be necessary to collect more examples for both opinions. The *headphones* and *hard disk* data are somewhere in the middle, which corresponds to the number of available examples.

8 Conclusions

The experiments with the five groups of textual data in natural language downloaded from the real world demonstrated that the procedure based on selecting a limited number of good patterns representing a certain opinion centered around a specific topic can provide acceptable results. The unlabeled samples are ranked according to their similarity with the patterns. It is up to a user how many of the most similar samples he or she takes for the future usage – with the increasing number of selected items the error of an inappropriate selection increases. The results also confirmed that more patterns increase the correctness of the selection. It is necessary to emphasize the fact that the goal was not to exactly separate classes – it would be a task for a classification procedure providing that all the training samples are labeled. A request from potential users is to select not too many (max. tens) 'good' items from hundreds, thousands, or much more. In such a case, manual labeling of all training examples would be too much expensive, if realizable. The continuing research is focused on more elaborate data preprocessing to improve the selection correctness for more and larger data collections. In addition, the study of dependence on different similarity algorithms is planned as well.

Acknowledgements. The research work published in this paper was supported by the Research program of Czech Ministry of Education VZ MSM 6215648904.

References

1. Srivastava, A.N., Sahami, M.: Text Miming: Classification, Clustering, and Applications. Chapmann and Hall/CRC, New York (2009)
2. Hroza, J., Žižka, J.: Mining Relevant Text Documents Using Ranking-Based k-NN Algorithms Trained by Only Positive Examples. In: Proceedings of Knowledge 2005, pp. 29–40. VŠB-Technical University, Ostrava (2005)
3. Hroza, J., Žižka, J.: Selecting Interesting Articles Using Their Similarity Based Only on Positive Examples. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 608–611. Springer, Heidelberg (2005)
4. Žižka, J., Hroza, J., Pouliquen, B., Ignat, C., Steinberger, R.: The Selection of Electronic Text Documents Supported by Only Positive Examples. In: Proceedings of the 8th International Conference on the Statistical Analysis of Textual Data, JADT 2006, Besançon, France, April 19-21, pp. 993–1002. Presses Universitaires de Franche-Comte (2006)
5. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2004, Seattle, Washington, August 22-25. ACM, New York (2004)
6. Amazon USA (March 2010), http://www.amazon.com
7. Amazon UK (March 2010), http://www.amazon.co.uk
8. MF Dnes (March 2010), http://mfdnes.newtonit.cz
9. ČSSD (March 2010), http://www.cssd.cz

Part III

Speech

"**Speech:** the expression of or the ability to express thoughts and feelings by articulate sounds: *he was born deaf and without the power of speech.*"
NODE (The New Oxford Dictionary of English), Oxford, OUP, 1998, page 1788, meaning 1.

Correlation Features and a Linear Transform Specific Reproducing Kernel

Andreas Beschorner and Dietrich Klakow

Spoken Language Systems, Saarland University, D-66123 Saarbrücken, Germany
{andreas.beschorner,dietrich.klakow}@lsv.uni-saarland.de
http://www.lsv.uni-saarland.de

Abstract. In this paper we introduce three ideas for phoneme classification: First, we derive the necessary steps to integrate linear transforms into the computation of reproducing kernels. This concept is not restricted to phoneme classification and can be applied to a wider range of research subjects. Second, in the context of support vector machine (SVM) classification, correlation features based on MFCC-vectors are proposed as a substitute for the common first and second derivatives, and the theory of the first part is applied to the new features. Third, an SVM structure in the spirit of phoneme states is introduced. Relative classification improvements of 40.67% compared to stacked MFCC features of equal dimension encourage further research in this direction.

Keywords: Correlation, Hilbert space, Reproducing kernel, Phoneme classification, SME structure.

1 Introduction

Established concepts like HMMs have long been in use for speech recognition and phoneme classification. In recent years systems have been influenced for example by generative models like GMMs [1], maximum a posteriori adaptive sequence estimation [2], discriminative methods [3] and along with the latter the theory of reproducing kernel Hilbert spaces (RKHS). In the context of reproducing kernels, sequence kernels [4] have been developed, capturing the non-static nature of speech or even modelling HMMs (see [5], pp. 430–436). Approaches like kernel combinations have been succesfully implemented in other fields of pattern recognition. [6,7] and [8] solve the SVM-optimization considering convex and linear combinations of kernels on (heterogenuous) compound feature vectors.

In our work, we pursue a new approach for phoneme classification. First, looking at feature computation, we show how to embed linear mappings into reproducing kernels. Second, we redesign SVM classification strategies and adopt concepts of subphonemes/ phoneme states without touching the kernel or the need to solve a modified optimization problem. The classification results encourage us to continue with research in this direction.

This paper is organized as follows. Subsequent to a brief review of the concepts of reproducing kernels and support vector machines, Section 3 shows how linear mappings can be embedded into the evaluation of reproducing kernels. These theoretical

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 235–242, 2010.

derivations in part motivate Section 4, in which we introduce MFCC-autocorrelation- and later cross-correlation features. Finally, we propose an SVM-based classification structuring approach (*SME* structure) utilizing a representation in the spirit of phoneme states. To get comparable information, we refrain from using kernel combinations for the new approach and compare classification results to experiments using stacked traditional MFCC-features (details are given in the respective section). The results of these experiments follow in section 5, and the paper closes with conclusions and perspectives in section 6.

2 Reproducing Kernels and SVMs

2.1 Reproducing Kernels

The concept of reproducing kernels is based on the fact that any Hilbert space $\mathcal{H}$ on a set X of complex-valued, bounded functionals endowed with an inner product $\langle\cdot,\cdot\rangle$ admits a mapping $k : X \times X \to \mathbb{C}$ such that for all $\mathbf{z} \in X$:

(1) $k(\cdot,\mathbf{z}) \in \mathcal{H}$
(2) $\forall f \in \mathcal{H} : f(\mathbf{z}) = \langle f, k(\cdot,\mathbf{z})\rangle$.

k is called a *reproducing kernel* and is unique within $\mathcal{H}$. It is easily verified ([9,10]) that reproducing kernels defined as such are positive semidefinite (psd). Conversely, for every psd $k : X \times X \to \mathbb{C}$ there exists exactly one $\mathcal{H} \subset \mathbb{C}$ wherein k is a reproducing kernel. Property (2) is called the *reproducing property*, as the kernel reproduces the evaluation of the functional $f \in \mathcal{H}$ using the Hilbert space's inner product. Given such a k, the factorization lemma ([11]) implies the existence of a Hilbert space $\mathcal{H}$ and a function $\Phi : X \to \mathcal{H}$ such that $k(\mathbf{x},\mathbf{z}) = \langle\Phi(\mathbf{x}), \Phi(\mathbf{z})\rangle$.

A reproducing kernel k thus allows us to replace costly computations of a mapping Φ by an inner product in $\mathcal{H}$. Well known examples are the linear kernel $k_l(\mathbf{x},\mathbf{z}) = \mathbf{x}^T\mathbf{z}$, the polynomial kernel $k_{p^d}(\mathbf{x},\mathbf{z}) = (\mathbf{x}^T\mathbf{z} + r)^d$ and the exponential kernel $k_e(\mathbf{x},\mathbf{z}) = \exp(-\gamma\,||\mathbf{x} - \mathbf{z}||^2)$. However, kernel functions are not generally restricted to numerical representations. Areas such as bioinformatics, data mining and part-of-speech-tagging in natural language processing, make frequent use of kernels defined on strings or more complex data structures like trees.

2.2 Support Vector Machines (SVMs)

Given a linear separable two-class dataset, SVMs compute a hyperplane $\mathbf{w}$ separating the two classes. The hyperplane is optimal in the sense that it has minimal margin amongst all hyperplanes separating the data, the margin being the distance from $\mathbf{w}$ to any (training) sample.

Let $\mathcal{H}$ be an N-dimensional Hilbert space on a set X, M be the number of samples. Writing $\mathbf{x} = (x_1 \cdots x_N)$ for $\mathbf{x} \in X$ and $n = 1,\ldots,N$, let $\mathbf{w} \in \mathcal{H}$ and $\mathbf{x}_1,\ldots,\mathbf{x}_M$ be vectors in X. With $b \in \mathbb{R}$ being the bias or offset, $\{\langle\mathbf{w},\mathbf{x}\rangle + b = 0 \,|\, \mathbf{x} \in \mathcal{H}\}$ is a subspace and hyperplane in $\mathcal{H}$ with normal vector $\mathbf{w}$. The dot product equals the length of the projection of either component onto the direction of the remaining one. Hence, the orientation of the hyperplane, $d(\mathbf{x}|\mathbf{w}) = \text{sgn}\,(\langle \mathbf{x}, \mathbf{w}\rangle + b)$, is a useful decision criterion. For target labels $y_m \in \{\pm 1\}, m = 1, \ldots, M$, the products $y_m \cdot d(\mathbf{x}|\mathbf{w})$ classify samples $\mathbf{x}$ into either class 1 or -1. The optimization problem of finding the hyperplane is subject to one constraint for each training sample: $y_m \cdot d(\mathbf{x}|\mathbf{w}) \geq 1, m = 1, \ldots, M$. To achieve better generalization, it has been proposed ([12] and, following them, [13]) to relax the constraints by introducing slack variables $\zeta_m \geq 1$, leading to soft margins. Using Lagrangian multipliers α_m to optimize under the constraints, the final dual form of the optimization problem,

$$\underset{\in \mathbb{R}^m}{\text{maximize}} \sum_{m=1}^{M} \alpha_m - \frac{1}{2} \sum_{m,l=1}^{M} y_m y_l \alpha_m \alpha_l \langle \mathbf{x}_m, \mathbf{x}_l \rangle \tag{1}$$
$$\text{s.t.} \quad \begin{cases} \langle \mathbf{y}, \boldsymbol{\alpha} \rangle = 0 \\ 0 \leq \boldsymbol{\alpha} \leq C \end{cases},$$

permits the substitution of the objective function's inner product by a kernel function k. The inequality $\boldsymbol{\alpha} \geq 0$ is to be understood elementwise, and the new decision function now is $d(\mathbf{x}|\mathbf{w}) = \sum_{m=1}^{M} \alpha_m y_m k(\mathbf{w}, \mathbf{x}_m) + b = 0$. As the dimension of the RKHS depends on the reproducing kernel, the data can be linearly separable in the RKHS even if this is not the case in the original space. For multiclass SVM cases, one-vs-one or one-vs-all strategies are commonly used. A thorough discussion is presented in [14] or [10].

3 Linear Mappings and Reproducing Kernels

Let us consider a continuous, linear mapping $T : X \to Y$ between two Banach spaces X, Y. A basic result from functional analysis is that the space $X \oplus Y$ with a norm given by $||(x, y)|| = \sqrt{||x||^2 + ||y||^2}, x \in X, y \in Y$ is again a Banach space. The graph $G(T) = \{(x, Tx)\,|\, x \in X\}$ of T is a closed subspace of $X \oplus Y$, the norm consequently being $||x||_T = \sqrt{||x||^2 + ||Tx||^2} \geq ||x||$. In Hilbert spaces, $G(T)$ as a closed subspace is itself a Hilbert space, its inner product defined on the concatenation of the components: Let $\mathcal{H}, \mathcal{H}_T$ be Hilbert spaces, $p, q \in \mathcal{H}$ and $T : \mathcal{H} \to \mathcal{H}_T$ with respective inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{H}_T}$. Then $G(T) \subset \mathcal{H} \oplus \mathcal{H}_T$, and

$$\langle (p, Tp), (q, Tq) \rangle_{G(T)} = \langle p, q \rangle_{\mathcal{H}} + \langle Tp, Tq \rangle_{\mathcal{H}_T}\,. \tag{2}$$

In this conetxt, a well-known theorem from Riesz ensures that for every bounded, linear continuous operator $T : \mathcal{H} \to \mathcal{H}_T$ between a finite dimensional Hilbert space $\mathcal{H}$ and a Hilbert space $\mathcal{H}_T$ there exists exactly one adjoint operator $T^* : \mathcal{H}_T \to \mathcal{H}$ such that for all $p \in \mathcal{H},\ q \in \mathcal{H}_T$ the equation $\langle Tp, q \rangle_{\mathcal{H}_T} = \langle p, T^* q \rangle_{\mathcal{H}}$ holds.

In our work we build on this theorem and, using the bilinearity of inner products in $\mathbb{R}$, recast equation (2) as follows:

$$\begin{aligned} \langle (p, Tp), (q, Tq) \rangle_{G(T)} &= \langle p, q \rangle_{\mathcal{H}} + \langle Tp, Tq \rangle_{\mathcal{H}_T} \\ &= \langle p, q \rangle_{\mathcal{H}} + \left\langle p, T^* T q \right\rangle_{\mathcal{H}} \\ &= \left\langle p, q + T^* T q \right\rangle_{\mathcal{H}} = \left\langle p, (I_{\mathcal{H}} + T^* T) q \right\rangle_{\mathcal{H}}\,, \end{aligned} \tag{3}$$

where $I_{\mathcal{H}}$ is the neutral element (that is, the identity matrix) of the endomorphisms of $\mathcal{H}$ and the last inner product is defined on $\mathcal{H} \times \mathcal{H}$. As $z^*(T^*T)z = (z^*T^*)(Tz) = (Tz)^*(Tz) \geq 0$, T^*T is psd and pd whenever the trace of T^*T does not equal zero. In this case, $(I_{\mathcal{H}} + T^*T)$ will be pd and a new reproducing kernel integrating T is given by $k_{T^*T}(p, q) = p^*(I_{\mathcal{H}} + T^*T)q$. Most important, the transform keeps vectors in the same space, allowing usual kernel combination techniques. In our experiments, we use this opportunity and apply kernel composition by inserting the new kernel into an exponential one.

The effect of T on the kernel can be controlled further by scaling the inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{H}_T}$ by numbers $w_{\mathcal{H}}$ and $w_{\mathcal{H}_T}$ respectively. Due to the linearity of inner products, this leads to

$$w_{\mathcal{H}} \langle p, q \rangle + w_{\mathcal{H}_T} \langle Tp, Tq \rangle = \left\langle p, (w_{\mathcal{H}} I_{\mathcal{H}} + w_{\mathcal{H}_T} T^*T) q \right\rangle_{\mathcal{H}}, \tag{4}$$

Note that $w_{\mathcal{H}}, w_{\mathcal{H}_T} > 0$ will ensure positive definiteness. For T unitary, equation (4) reduces to scaling q, as

$$\langle (p, Tp), (q, Tq) \rangle_{G(T)} = \left\langle p, \left((w_{\mathcal{H}} + w_{\mathcal{H}_T}) I_{\mathcal{H}} \right) q \right\rangle_{\mathcal{H}}. \tag{5}$$

4 MFCC-Correlation Features

In this section we introduce new features consisting of usual MFCC-vectors plus its correlation with adjacent vectors, the latter replacing the widely used Δ and $\Delta\Delta$. In our setting, both parts of such vectors are convex combined via reproducing kernels during training and classification. While different components of MFCC feature vectors (and thus the different frequency subbands they represent) are decorrelated via a discrete cosine transform in the process of their computation, correlation remains within sequences of the same component. To keep the number of features reasonable, we only consider immediately adjacent MFCC features of appropriate length L.

Formalizing this, let $m^l_{n-1}, m^l_n, m^l_{n+1}, m^l_{n+2}, n \in \mathbb{N}$ be a sequence of adjacent MFCC vectors, where $l = 1, \ldots, L$ references a component of the vectors and the subindex n the speech frame the MFCC features were computed from.

Using $\times$ to indicate cross correlation and forming two vectors, each of length three, of the same components of adjacent vectors, we get L cross correlation vectors

$$\tilde{m}_l = \left(m^l_{n-1}, m^l_n, m^l_{n+1} \right) \times \left(m^l_n, m^l_{n+1}, m^l_{n+2} \right).$$

Normalising and stacking the $\tilde{m}_l$ finalizes the computation of the autocorrelation feature vector $\left(\tilde{m}^1, \ldots, \tilde{m}^L \right)$.

4.1 Linearization and a Phoneme State-Like Approach

Given a fixed vector x of finite length, correlation with any finite vector y is a transform linear in y. However, this is not true for the autocorrelation we just discussed. In order to apply the theory presented in Section 3, we therefore linearize the process as follows.

A phoneme state kind of representation is simulated by splitting (training) samples of length s into start- and endsection (S and E respectively) of length $s \div 3$ and a middle section (M) of length $(s \div 3) + (s \mod 3)$. For each specific phoneme/class, we group those subfeatures, compute their averages and denote these centers by x_S, x_M and x_E. In a first approach we allow seven *SME*-based states: *SSS, SSM, SMM, MMM, MME, MEE, EEE*. They represent the position in a phoneme, and training sets are built based on this segmentation. Using the new vectors, we switch from auto- to crosscorrelation, applying the theory following immediately. Figure 1 illustrates, which 3-vector sequences of a phoneme sample contribute to which training set.

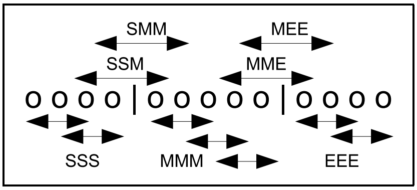


Fig. 1. Example of a phoneme sample comprised of 13 MFCC-vectors/ frames and its split into *SME*-based parts. Two vectorsequences are used for the classes *SSS* and *EEE*, three for *MMM* and one each for *SSM, SMM, MME* and *MEE*.

In comparison to autocorrelation, the computation now depends on the phoneme state. Considering a vector coordinate $1 \le l \le L$, the linear mapping is computed via $\left(x^l_{qs}, x^l_{qm}, x^l_{qe}\right) \times \left(m^l_n, m^l_{n+1}, m^l_{n+2}\right)$, where $qs, qm, qe \in \{S, M, E\}$. In matrix form (omitting the index l for clarity), the linear transform equals

$$T = \begin{pmatrix} 0 & 0 & x_{qs} \\ 0 & x_{qs} & x_{qm} \\ x_{qs} & x_{qm} & x_{qe} \\ x_{qm} & x_{qe} & 0 \\ x_{qe} & 0 & 0 \end{pmatrix}$$

and along with this result we substitute

$$T^*T = \begin{pmatrix} x^2_{qs} + x^2_{qm} + x^2_{qe} & x_{qs}x_{qm} + x_{qm}x_{qe} & x_{qs}x_{qe} \\ x_{qs}x_{qm} + x_{qm}x_{qe} & x^2_{qs} + x^2_{qm} + x^2_{qe} & x_{qs}x_{qm} + x_{qm}x_{qe} \\ x_{qs}x_{qe} & x_{qs}x_{qm} + x_{qm}x_{qe} & x^2_{qs} + x^2_{qm} + x^2_{qe} \end{pmatrix}$$

in equation (4). If, for instance, the state in question is SSM, x_{qs} and x_{qm} take values from the cluster center vector x_s, whereas x_{qe} is set to the respective coordinate of x_m. Following the derivation of the previous section, T^*T will be pd and hence k_{T^*T} a reproducing kernel whenever $x^2_{qs} + x^2_{qm} + x^2_{qe} \neq 0$.

5 Experimental Results

We present results from two sets of multiclass classification results. Section 5.1 refers to the autocorrelation features introduced in Section 4 and Part 5.2 depicts the experiments of the cross correlation setup described in Section 4.1. The features were extracted via HTK3.3 with a framesize of 25ms and an overlap of 10ms. Training and test were performed on the eleven most frequent phonemes *aa, ae, ay, eh, ey, ih, ix, iy, n, s, z* of the TIMIT dataset using a modified version of *svmlight* [15]. If not mentioned otherwise, *svmlight*-parameters remained unchanged. Also, parameters for SVMs trained on the new vectors were not optimized but chosen due to results from partially rough grid tests. Evaluating on finer grids and, in the case of kernel combination, solving the convex kernel combination SVM optimization problem will very likely improve results further.

5.1 Autocorrelation Features

For the SVM-classification baseline we use standard MFCC features consisting of 13 values plus Δ and $\Delta\Delta$ and a single exponential kernel. γ is set to 0.001, a quasi-optimal value determined by a rough grid search. This is compared to SVM-classification using the autocorrelation features. Two exponential kernels on the two parts forming the vector – the 13 basic values and the 26 autocorrelation values – are convex combined with unchanged $\gamma = 0.001$. Using a convex weighting $wk_e^{mfcc} + (1-w)k_e^{corr}$ and a 1-vs-1 setup, we perform a rough first evaluation for $w = 0.05, 0.10, \ldots, 0.95$. Table 1 illustrates the results for $w = 0.95$.

Table 1. Results of classification error rates for two setups: Features of size 39 (13 MFCC-values plus Δ and $\Delta\Delta$) using one RBF-kernel and features of size 39 (13 MFCC-values plus autocorrelation values) using a kernel combination. The latter produces a slight drop of the cumulative average classification error for all w. $w = 0.95$ gives the best results so far: A relative classification improvement of about 7.5%. For two classes, *ix* and *ay*, classification abates, while elsewhere it improves.

phoneme	*aa*	*ae*	*ay*	*eh*	*ey*	*ih*	*ix*	*iy*	*n*	*s*	*z*	**avg.**
MFCC39	38.7	73.4	63.9	64.2	65.8	94.1	25.2	22.9	44.8	12.4	88.4	**54.6**
MFCC13+corr	32.6	66.8	70.1	61.2	64.2	89.9	42.3	26.1	26.7	5.8	77.9	**51.2**

5.2 Crosscorrelation Features and Phoneme State Simulation

As the new correlation features extend over three frames, comparison to single-frame MFCC-features is improper due to the difference in the amount of information. We thus consider 3-vector sequences of standard 13-dimensional MFCC-features (sMfcc) without Δ and $\Delta\Delta$, resulting in comparable feature vectors of equal dimension. To get a first impression of the quality of this new approach, we use a single exponential kernel. The γ-parameter is again selected due to a rough grid search and set to 0.0001 for the sMfcc-SVMs and to 0.00001 for the *SME*-SVMs and let $w_{\mathcal{H}_T} = w_{\mathcal{H}} = 0.5$. Table 2 illustrates the strong raise in the recognition rates.

Table 2. Recognition rates of *SME*-based classification compared to sMfcc features. Even phonemes like *ih* and *ix* that are hard to tell apart and often merged in experimental setups ([3], e.g.) are separated relatively well. The overall relative recognition gain is approximately 40.67%.

	sMfcc	SSS	SSM	SMM	MMM	MME	MEE	EEE	**SME-avg.**
aa	**69.26**	75.53	87.72	92.01	94.08	95.72	95.91	91.82	**90.40**
ae	**59.79**	94.27	83.54	83.71	92.66	79.65	79.96	88.76	**86.08**
ay	**52.22**	83.30	86.23	91.77	96.61	91.94	80.72	82.66	**87.60**
eh	**44.22**	62.54	88.44	91.44	92.23	90.95	87.68	65.58	**82.69**
ey	**56.39**	82.77	81.22	83.04	92.38	83.64	84.10	90.47	**85.37**
ih	**37.82**	79.72	77.50	71.74	75.85	73.40	83.07	90.41	**78.81**
ix	**47.63**	40.17	84.30	94.16	86.05	95.87	92.02	21.12	**73.38**
iy	**77.44**	96.21	96.66	96.72	97.68	96.40	96.03	91.32	**95.86**
n	**88.63**	95.71	97.96	98.23	97.79	98.35	97.36	88.57	**96.28**
s	**88.43**	98.59	96.35	94.89	96.15	91.88	92.36	99.37	**95.66**
z	**42.50**	77.45	87.39	79.38	48.61	66.60	63.04	21.01	**63.35**
avg.	**60.39**								**84.95**

6 Conclusion, Discussion and Perspectives

In this paper we have introduced correlation features computed from adjacent frames of MFCC-vectors of phonemes. We derived a kernel that integrates a linear mapping and proposed a new SVM-structure akin to the idea of phoneme states. Phoneme classification experiments show great improvements and encourage further research.

Clearly, not all phonemes are adequately represented by decomposition into all of the seven states or do not even deliver samples for the *SME*-trainingsets due to their size. Entries in Table 2 like the EEE results for *ix* and *z* reflect this situation. Hence, individual setups are currently evaluated. One step in classification will be intra-class evaluations, choosing the one (or even two) classes with best recognition rates for further between-class classification. We are positive that this will again improve the results. Rebalancing the *SME*-split (section 4.1) and in line with this the sizes of the SVMs' training sets can further attribute to the recognition performance.

Finally, the definition of a graph also holds for operators, and the respective equation – equation (2) in Section 3 – can be recast in a similar way. In this context, the theory presented here becomes interesting for functions known to be reproducing kernels of for instance Sobolev- and Hardy spaces. For both operators and mappings, care must however be taken that $\langle Tf, Tg \rangle$ remains an inner product. Differentiation for instance annihilates its definitness.

References

1. Rodríguez, E., et al.: Speech/Speaker Recognition Using a HMM/GMM Hybrid Model. LNCS, vol. 1206, pp. 227–234. Springer, Heidelberg (1997)
2. Chakrabartty, S., Cauwenberghs, G.: Forward-Decoding Kernel-Based Phone Sequence Recognition. Adv. Neural Information Processing Systems (2002)

3. Smith, N., Gales, M.J.F.: Using SVMs and Discriminative Models for Speech Recognition. In: IEEE International Conference on Accoustic Speech and Signal Processing, vol. 1, pp. 77–80 (2002)
4. Campbell, W.M.: A Sequence Kernel and its Application to Speaker Recognition. In: Neural Information Processing Systems, vol. 14, pp. 1157–1163 (2001)
5. Shawe-Taylor, J., Chrstianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
6. Lanckriet, G.R.G., et al.: Learning the Kernel Matrix with Semidefinite Programming. Journal of Machine Learning 5, 27–72 (2004)
7. Lanckriet, G.R.G., et al.: A Statistical Framework for Genomic Data Fusion. Bioinformatics 20, 2626–2635 (2004)
8. Hirokai, T., et al.: Simple but Effective Methods for Combining Kernels in Computational Biology. RIVF, 71–78 (2008)
9. Aronszajn, N.: Theory of Reproducing Kernels. Transactions of the American Mathematical Society 68, 337–404 (1950)
10. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
11. Agler, J., McCarthy, K.: Pick Interpolation and Hilbert Functions Spaces. Graduate Studies in Mathematics, vol. 44. American Mathematical Society, Providence (2002)
12. Bennet, K., Magasarian, O.: Robust Linear Programming Discriminaion of Two Linearly Inseparable Sets. Optimization Methods and Software 1, 22–34 (1992)
13. Cortes, C., Vapnik, V.: Support Vector Machines. Machine Learning 20, 273–297 (1995)
14. Shigoe, A.: Support Vector Machines for Pattern Classification. In: Advances in Pattern Recognition, Springer, Heidelberg (2005)
15. Joachims, T.: Making Large-Scale SVM Learning Practical. Advances in Kernel Methods – Support Vector Learning (1999)

Automatic Detection and Evaluation of Edentulous Speakers with Insufficient Dentures

Tobias Bocklet[1,2], Florian Hönig[1], Tino Haderlein[1,3],
Florian Stelzle[2], Christian Knipfer[2], and Elmar Nöth[1]

[1] Universität Erlangen-Nürnberg, Lehrstuhl für Mustererkennung (Informatik 5),
Martensstraße 3, 91058 Erlangen, Germany
tobias.bocklet@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de
[2] Universität Erlangen-Nürnberg, Mund-, Kiefer- und Gesichtschirurgische Klinik,
Glückstraße 11, 91054 Erlangen, Germany
[3] Universität Erlangen-Nürnberg, Abteilung für Phoniatrie und Pädaudiologie,
Bohlenplatz 21, 91054 Erlangen, Germany

Abstract. Dental rehabilitation by complete dentures is a state-of-the-art approach to improve functional aspects of the oral cavity of edentulous patients. It is important to assure that these dentures have a sufficient fit. We introduce a dataset of 13 edentulous patients that have been recorded with and without complete dentures in situ. These patients have been rated an insufficient fit of their dentures, so that additional (sufficient) dentures and additional speech recordings have been prepared. In this paper we show that sufficient dentures increase the performance of an ASR system by ca. 27 %. Based on these results, we present and discuss three different systems that automatically determine whether the dentures of an edentulous person have a sufficient fit or not. The system with the best performance models the recordings by GMMs and uses the mean vectors of these GMMs as features in an SVM. With this system we were able to achieve a recognition rate of 80 %.

Keywords: speech recognition, user modeling, assistive technology, applied system.

1 Introduction

A complete loss of teeth can cause persistent speech disorders by altering dental articulation areas. This severely reduces the quality of speech [1] and even the recognition rate of an automatic speech recognition system [2,3]. Removable complete dentures partly solve these problems [4].

Dental rehabilitation of edentulous patients by complete dentures improves not only esthetic and functional aspects, e.g., mastication of food, but also the speech quality. However, complete dentures restrict the flexibility of the tongue, narrow the oral cavity and alter the articulation areas of the palate and teeth. This is even amplified when the fit of the dentures is not sufficient.

In this paper we first introduce a dataset of 13 edentulous speakers that have been recorded with and without complete dentures in situ. After a later examination these

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 243–250, 2010.

dentures have been rated not to have a perfect fit, i.e., an insufficient fit. This aspect required a new preparation of a complete denture for each of these 13 patients. So finally three recordings where available for each of these 13 patients: one without complete dentures, one with insufficient dentures and one with sufficient dentures.

In this paper we first examined in which way these three types of recordings affect the performance of an automatic speech recognition system. The main goal of this paper was to create a system that is able to detect on a spoken text whether the complete dentures have a sufficient fit or not. We describe three different systems to achieve this goal: One system takes the word accuracy results of a speech recognizer and uses this single value as feature. The second system calculates the distance between the spoken text of an edentulous patient and a reference speaker by Dynamic Time Warping (DTW). These distances are then used as features for a classifier. The other system models each speaker by Gaussian Mixture Models (GMMs) and uses the concatenated mean vectors of these GMMs as input vector for a classifier.

The outline of this paper is as follows: Section 2 describes the dataset we used, Section 3 describes our recognition system and the two systems we used for detecting insufficient dentures. The results are presented and discussed in Section 4. We finish with a summary in Section 5.

2 Dataset

The original dataset was first introduced in [2]. In contains 28 edentulous, i.e. toothless, patients. Their average age was 64 ± 10 years. Only patients who wore removable complete dentures where chosen to participate in this study. Only patients wearing their dentures for at least one month where chosen in order to ensure a patient habituation to their new dentures. All patients were native German speakers who were asked to speak standard German while being recorded. None of the patients had speech disorders caused by medical problems others than dental or any report of hearing impairment.

After a later examination by a senior dentist 13 patients had been rated to have complete dentures with an insufficient fit. The complete denture was rated as insufficient if one of the following seven parameters was rated as insufficient/ not correct:

- Absence of pain concerning the chewing muscles, the soft and hard tissue in functional and non- functional situations
- Absence of variances of the soft tissue like redness or ulcer
- Ability to chew and swallow without restrictions
- Balanced occlusion relationship under function
- An interocclusal distance of 2 mm in a physiologic rest position
- Excellent fit proven by a soft pattern
- Patient satisfaction

For these 13 patients, additional dentures have been produced which have been rated a sufficient fit afterwards. Finally, the patients read the text “Der Nordwind und die Sonne” (NWS) three different times: One time without their complete dentures in situ, one time while wearing their insufficient dentures and one time wearing their sufficient dentures. The NWS text is a phonetically balanced text with 108 words (71 disjunctive) which is used in German speaking countries in speech evaluation and therapy. The speech data was sampled with a frequency of 16 kHz and an amplitude resolution of 16 bit.

3 Methods

In this section we first describe our feature front-end processing. These features are used in our ASR system (Section 3.2) and in two of the three denture classification systems (see Section 3.3).

3.1 Feature Extraction

As features we use the well-known Mel-frequency cepstrum coefficients (MFCCs). These features perform a short-time analysis of the speech signal. Therefore a Hamming window with a length of 16 ms and a frame rate of 10 ms is applied to the signal. The filter-bank for the Mel spectrum with 25 triangle filters is calculated afterwards. The cepstral coefficients are computed by an inverse discrete cosine transform of the logarithmic Mel spectrum. For each frame a 24-dimensional feature vector is created. It contains the short-time energy, 11 Mel-frequency cepstral coefficients and their first-order derivatives approximated by the slope of a regression over 5 consecutive frames.

3.2 Automatic Speech Recognition System

The speech recognition system used for some of the experiments was developed at the Chair of Pattern Recognition in Erlangen [5]. The system is based on semi-continuous Hidden Markov Models (HMM). It can model phones in a context as large as statistically useful and thus forms the so-called polyphones, a generalization of the well-known bi- or triphones. The HMMs for each polyphone have three to four states with a codebook containing 500 Gaussian mixtures with full covariance matrices. The feature front-end of our ASR system is described in Section 3.1.

Our ASR system was trained on German dialogues from the VERBMOBIL project [6]. The data was recorded with a close-talking microphone and a sampling frequency of 16 kHz. It was quantized with 16 bit. About 80% of the 578 training speakers (304 male, 274 female) were between 20 and 29 years old, less than 10% were over 40. 11,714 utterances (257,810 words) of the VERBMOBIL-German data (12,030 utterances, 263,633 words, 27.7 hours of speech) were used for training and 48 (1,042 words) for the validation set, i.e. the corpus partitions were the same as in [5].

The vocabulary of known words of the ASR system was changed to the 71 words of the NWS text. The word accuracy (WA) and the word recognition rate (WR) are used as basic measures. They are computed from the comparison between the recognized word sequence and the reference text consisting of the $n_{\text{all}} = 108$ words of the read text. With the number of words that were wrongly substituted (n_{sub}), deleted (n_{del}) and inserted (n_{ins}) by the recognizer, the word accuracy in percent is given as

$$\text{WA} = [1 - (n_{\text{sub}} + n_{\text{del}} + n_{\text{ins}})/n_{\text{all}}] \cdot 100 \qquad (1)$$

Only a unigram language model was used in our recognition experiments in order to put more emphasis onto the acoustic models.

3.3 Classification Systems

In this section we describe the three systems we proposed to detect insufficient dentures automatically. Our baseline system is based on the word accuracy rates of the speech recognizer. The second system uses dynamic time warping (DTW) and calculates the distance between a speaker with dentures and a reference speaker. The third system models each speaker with Gaussian Mixture Models (GMMs) and uses the GMMs as meta-features within a classification system.

System with word accuracy features. Our baseline system uses a one dimensional feature vector. This feature represents the WA of the speech recognition system as single feature. The system is motivated by the fact, that we achieved different mean WA values for the three subsets of our dataset (see Section 4.1). This one-dimensional feature vector is then classified by a Support Vector Machine (SVM) [7].

System with Dynamic Time Warping. In a second approach, we use Dynamic Time Warping (DTW) [8] to extract a feature vector coding distances between the recording of the speaker with dentures to be assessed (*test speaker*) and a (non-pathologic) reference realization of the same text in the MFCC feature space. This distance feature vector is then classified by an SVM.

More precisely, we compute the series of acoustic feature vectors for the two realizations of the NWS – text (mean-normalized MFCCs, augmented by the first derivatives), and compute a mapping between them such that the accumulated (squared) distance between corresponding feature vectors plus the costs incurred by insertions and deletions is minimal (we use a suitable weighting of energy (first cepstral coefficient), the derivatives and insertions/deletion penalties). This mapping relates corresponding phonemes of the test and reference speaker very reliably since the recordings contain the same words; where the test speaker inserts or deletes words, according large insertion/deletion penalties occur. Substituted words or phonemes are usually reflected by both larger acoustic distances and higher insertion/deletions penalties. We then construct a feature vector for classification that contains for each frame j of the reference sound the average distance to those acoustic feature vectors of the test speaker that frame j is assigned to, and the penalties for insertions and deletions. Thus, we obtain a feature vector of fixed size (which is necessary for classification) that details the test speaker's deviations from the reference speaker over time. It is left to the classifier to concentrate on those parts of the recording (i.e. elements of the distance feature vector) that are important for classification performance. This strategy is much more promising than e.g. just using the average distance between assigned feature vectors.

System with Gaussian Mixture Models. The third system is based on Gaussian Mixture Models (GMMs). We used this system for the task of intelligibility assessment on children with Cleft Lip and Palate [9] and on speakers with partial laryngectomy [10]. We base our evaluation only on a shift of the mean vectors from a Universal

Background Model (UBM) to the mean vectors of a speaker GMM. GMMs model the acoustic features, and with this the acoustic space, of a specific recording. A GMM (λ) contains M unimodal Gaussian densities. Each density represents a different acoustic area of the feature space:

$$p(\vec{c}|\vec{\lambda}) = \sum_{i=1}^{M} \omega_i \, p_i(\vec{c}|\vec{\mu}_i, \vec{\Sigma}_i) = \quad (2)$$

$$= \sum_{i=1}^{M} \omega_i \cdot \frac{1}{(2\pi)^{D/2}|\vec{\Sigma}_i|^{(1/2)}} e^{-(1/2)(\vec{c}-\vec{\mu}_i)^T \vec{\Sigma}_i^{-1}(\vec{c}-\vec{\mu}_i)}, \quad (3)$$

The idea is now to train GMMs, extract the mean vectors of the GMMs for each recording of one speaker, i.e., with sufficient and with insufficient dentures, concatenate them and classify these vectors with an SVM.

After feature extraction (Section 3.1) a UBM is created on a dataset of healthy speakers who also read the NWS text. This is achieved by using 5 iterations of the EM algorithm [11]. Beginning with this UBM, a speaker-dependent GMM is built by MAP adaptation [12]. The MAP adaptation takes the UBM as an initial model and adapts the statistics to the acoustic features of a specific speaker in a single iteration step. These new densities are combined with the UBM statistics afterwards. Finally, a GMM λ is created for each recording. The components of each GMM are concatenated to a GMM-based supervector. These supervectors can be regarded as a mapping from the acoustics of a recording, i.e., MFCCs, to a higher-dimensional feature vector which represents the acoustic characteristics of this recording. Since we are dealing with two types of recordings, i.e., recordings of persons with sufficient and insufficient dentures, we expect the acoustic characteristics of these two types being different and expect them to be modeled by these GMM supervectors.

4 Experiments and Results

The results of this paper can be split into two different parts: First we describe the WA results on the three different types of recordings of our 13 patients. These different types are recordings were the persons did not wear their dentures, recordings were they wear insufficient dentures, and recordings with sufficient dentures in situ.

In the second part of this section we present and discuss the results on recognizing whether a recording was performed with sufficient or insufficient dentures.

4.1 Speech Recognition

The automatically computed WA differed for the three different subsets (see Table 1). The WA on recordings without dentures (60.06 %) was lower than recordings with insufficient dentures (64.35 %). In the case of sufficient dentures a WA of 70.91 % was measured. This is an improvement by 18 % compared to the recordings without dentures and 10 % compared to the recordings with insufficient dentures. The standard deviation on recordings without any dentures was ±10.35. The value decreased to ±9.64

Table 1. Word accuracy (WA) result, according standard deviation and minimum/maximum WA value for the three different subsets: Without wearing dentures, with insufficient dentures, with sufficient dentures

dataset	mean WA	std. dev	min WA	max WA
without	60.06	± 10.35	39.48	77.78
insufficient	64.35	± 9.64	45.37	80.56
sufficient	70.91	± 6.04	57.51	80.56

and ±6.04 when wearing insufficient dentures or sufficient dentures, respectively. This is an improvement of 37 % when comparing the values of insufficient and sufficient dentures. This improvement is also visible when focusing on the minimum word accuracy values. Insufficient dentures improved the minimal WA from 39.48 % to 45.37 %. This is an improvement by 15 %. Wearing sufficient dentures again improved the value by 27 % to a WA of 57.51 %.

4.2 Denture Classification

The two-class problem of identifying whether the complete dentures have a sufficient fit or not was handled by three different systems. Since it was not the goal of this paper to select the classifier with the best performance, we selected an SVM to be used for each experiment. The dataset used in this paper had a very limited number of samples. It contains 13 speakers, with two recordings for each speaker; one time with insufficient dentures and one time with sufficient dentures. To deal with that problem, we performed our experiments in a cross-validation with leave-one-speaker-out manner.

System number one uses a one-dimensional feature vector, i.e., the word accuracy of the recognizer. With this baseline system, a recognition result of $61.\dot{5}$ % was achieved for the two-class problem; 7 recordings of sufficient dentures have been classified correctly, and 9 recordings with insufficient dentures have been classified correctly.

The second system uses the DTW distances as features for an SVM classification. These distances have been calculated with respect to the recordings of a reference speaker without dentures. The length of the feature vector was $2,314$. 8 of 13 insufficient recordings have been identified correctly and 11 recordings have been correctly identified as insufficient. This system achieved a recognition result of 73.1 %. This is a relative improvement of 19 % compared to the baseline system.

The third system uses the mean vectors of a 128-dimensional GMM as features. The dimension of this GMM supervector is $24 * 128 = 3,072$. Again, an SVM was used for classification. The number of correctly classified recordings with sufficient dentures was 10; 11 recordings of insufficient dentures have been classified correctly. This sums up to a recall of 80.1 %. Compared to the baseline system the GMM system achieved an significant improvement ($p < 0.1$) of 30 %. Compared to the DTW-based system the system achieved a relative improvement of 9.5 %.

Table 2. Recognition results of the three different systems on the problem of detecting insufficient dentures

system	feature dim	corr. sufficient	corr. insufficient	recognition result
WA	1	7	9	61.5 %
DTW	2,314	8	11	73.1 %
GMM	3,072	10	11	80.1 %

5 Summary

In this paper we performed ASR experiments on a dataset of 13 edentulous patients. Complete dentures have been produced for these speakers. since these dentures have been rated an insufficient fit, new dentures have been created for these speakers. So for each speaker three recordings have been available. We performed ASR experiments on these data that showed an improvement of the mean WA of 10 % between recordings without any dentures and recordings with insufficient dentures. Sufficient dentures improved these results by another 18 %. The second task of this paper was the automatic identification of incomplete dentures. Therefore we compared the recognition results of three different system: one system used only the WA result as feature, a second system used the DTW distances w.r.t. to a reference speaker and the third system used the mean vectors of 128 dimensional GMMs as features. We achieved a recognition rate of 80 % for this two-class problem.

Acknowledgement

This work was supported by the Wilhelm Sander-Foundation, Germany (AZ: 2007.100.1). The authors are responsible for the content of this article.

References

1. Ichikawa, J., Komoda, J., Horiuchi, M., Matsumoto, N.: Influence of Alterations in the Oral Environment on Speech Production. Journal Of Oral Rehabilitation 22, 295–299 (1995)
2. Haderlein, T., Bocklet, T., Maier, A., Nöth, E., Knipfer, C., Stelzle, F.: Objective vs. Subjective Evaluation of Speakers with and without Complete Dentures. In: Matousek, V., Mautner, P. (eds.) Proc. Text, Speech and Dialogue; 12th International Conference, Berlin. LNCS (LNAI), vol. 1, pp. 170–177 (2009)
3. Stelzle, F., Uginovic, B., Knipfer, C., Bocklet, T., Nöth, E., Schuster, M., Eitner, S., Seiss, M., Nkenke, E.: Automatic, Computer-Based Speech Assessment on Edentulous Patients with and without Complete Dentures – Preliminary Results. Journal Of Oral Rehabilitation 37, 209–216 (2010)
4. Tanaka, H.: Speech Patterns of Edentulous Patients and Morphology of the Palate in Relation to Phonetics. The Journal of Prosthetic Dentistry 29, 16–28 (1973)
5. Stemmer, G.: Modeling Variability in Speech Recognition. Studien zur Mustererkennung, vol. 19. Logos Verlag, Berlin (2005)
6. Wahlster, W. (ed.): Verbmobil: Foundations of Speech-to-Speech Translation. Springer, Berlin (2000)

7. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
8. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26(1), 43–49 (1978)
9. Bocklet, T., Maier, A., Riedhammer, K., Nöth, E.: Towards a Language-independent Intelligibility Assessment of Children with Cleft Lip and Palate. In: Workshop on Child, Computer, and Interaction 2009, New York (2009)
10. Bocklet, T., Haderlein, T., Hönig, F., Rosanowski, F., Nöth, E.: Evaluation And Assessment Of Speech Intelligibility On Pathologic Voices Based Upon Acoustic Speaker Models. In: Proceedings of the 3rd Advanced Voice Function Assessment International Workshop, Madrid, pp. 89–92 (2009)
11. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B (Methodological) 39(1), 1–38 (1977)
12. Gauvain, J., Lee, C.: Maximum A-Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. IEEE Transactions on Speech and Audio Processing 2, 291–298 (1994)

Diagnostics for Debugging Speech Recognition Systems

Miloš Cerňak

Institute of Informatics, Slovak Academy of Sciences,
Dúbravská cesta 9, 045 07 Bratislava
milos.cernak@savba.sk

Abstract. Modern speech recognition applications are becoming very complex program packages. To understand the error behaviour of the ASR systems, a special diagnosis–a procedure or a tool—is needed. Many ASR users and developers have developed their own expert diagnostic rules that can be successfully applied to a system. There are also several explicit approaches in the literature for determining the problems related to application errors. The approaches are based on error and ablative analyses of the ASR components, with a blame assignment to a problematic component. The disadvantage of those methods is that they are either quite time-consuming to acquire expert diagnostic knowledge, or that they offer very coarse-grained localization of a problematic ASR part. This paper proposes fine-grained diagnostics for debugging ASR by applying a program-spectra based failure localization, and it localizes directly a part of ASR implementation. We designed a toy experiment with diagnostic database OLLO to show that our method is very easy to use and that it provides a good localization accuracy. Because it is not able to localize all the errors, an issue that we discuss in the discussion, we recommend to use it with other coarse-grained localization methods for a complex ASR diagnosis.

Keywords: automatic speech recognition, fault diagnosis.

1 Introduction

With increasing complexity of the ASR systems (advanced algorithms, multi- and many-processors implementations and so on), effective diagnostics has to be proposed. It is obvious mainly in the commercial environment where the time for error analysis is often a critical parameter and a fast solution, an automatic or semi-automatic method, is required. Academic groups can profit from such diagnostics as well, as it could improve understanding of the error behaviour of the system, which could result in better tuning of the systems.

There are several approaches in the literature that propose the diagnostic methods. They try to answer the question: How much error is attributable to each of the components? **Error analysis** tries to explain the difference between current performance and perfect performance. As a ground-truth a forced alignment is used, and then an analysis is done on error regions with respect to partial acoustic and language modeling scores. In [1] a diagram, a sequence of tests applied to the error region produced by an alignment of recognized and forced aligned phones, predicts a category to each error region. Front-end, language modeling, and the search belong to the main error

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 251–258, 2010.

categories, and this is directly linked to a concrete component of the ASR in the test. A similar approach was proposed in [2], where the error-causing components were exactly the three mentioned above. Binary decision trees are mostly employed as error predictors [3], this being a popular approach not only in ASR (an example is [4]). **Ablative analysis**, on the contrary, tries to explain the difference between a much poorer baseline performance and current performance. A good example can be found in [5]. The authors present a system with a poor baseline with an average WER of 51.1%, and with application of more components in a pipeline or a component combinations, they achieved an average WER of 32.0%, with a calculated contribution of each component to the error rate. Both error and ablative analyses offer coarse-grained localization of the error-causing components.

To achieve fine-grained localization we can either partition existing components to sub-components, which might be a laborious task without any effect on the system performance, or we can directly analyse the implementation of the components. In this paper we explore the second approach, focusing on program-spectra based error localization [6]. The key idea is linking dynamic program events (program spectra) with erroneous program behaviour at the level of code lines, and localize the lines that match error indicators most closely. Diagnostic accuracy is then computed to rate the finding for a human examiner.

The remainder of the paper is structured as follows. In the next Section 2 we introduce software fault diagnosis, which we apply to a special test scenario in the experimental part of the paper in Section 3. Finally in Section 4 we discuss pros and cons of the approach.

2 Software Fault Diagnosis

Model-based diagnosis is a central point of the fault diagnosis theory. The model of a system serves as a definition of its intended behavior, and may contain additional information about its composition and operation. While model-based diagnosis has successfully been applied for diagnosing complex mechanical systems, its application to software has proven to be difficult. Spectra-based fault localization has been proposed recently for complex software system [6].

Central to the discussion of diagnosis are the notions of failure and fault. A failure is a discrepancy between expected and observed behavior, and a fault is a property of the system that causes such a discrepancy. Fault localization is a task of finding a place in the program, which should be responsible for a failure. The fault in terms of software can be considered in the worst case as a bug in the program code.

A program spectrum is a collection of data that provides a specific view on the dynamic behavior of software. This data is collected at run-time using program profiling tools. In this paper we work with code coverage spectra that indicate whether or not a block of code was executed in a particular run. By the block we mean a C/C++ language statement (but the method is also applicable for different languages). The tutorial [6] gives an excellent overview of the program analysis technique. Once the potential spectra-based fault has been localized, the assessment requires understanding of the correlation between localization and program behavior.

$$\begin{array}{ccc} & P\text{ parts} & \text{errors} \\ R\text{ spectra} & \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ x_{21} & x_{22} & \cdots & x_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{R1} & x_{R2} & \cdots & x_{RP} \end{bmatrix} & \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_R \end{bmatrix} \\ & s_1\ s_2\ \ldots\ s_P & \end{array}$$

Fig. 1. $R \times P$ binary program spectra matrix and $R \times 1$ error vector. Similarity coefficients $s_1, s_2, \ldots, s_P$ are calculated for each column of the spectra matrix.

2.1 Fault Localization

A test set for regression testing of a speech recognizer consists of R utterances. The hit spectra of R ASR runs constitute a binary matrix, whose columns correspond to P different blocks of the program code. The information about failure detection (misrecognition) constitutes another column vector, the error vector (see Fig. 1). This vector can be construed as representing error indicators. Fault localization essentially consists of identifying which column vectors resemble the error vector most [6].

Similarity coefficients are used as quantifiers of resemblances between the columns of program spectra matrices. The coefficients are used to express the similarity s_j of the column j and the error vector. As an example, below are three different similarity coefficients:

- The Jaccard similarity coefficient:

$$s_J(j) = \frac{a_{11}(j)}{a_{11}(j) + a_{01}(j) + a_{10}(j)} \tag{1}$$

- The Tarantula coefficient:

$$s_T(j) = \frac{\frac{a_{11}(j)}{a_{11}(j)+a_{01}(j)}}{\frac{a_{11}(j)}{a_{11}(j)+a_{01}(j)} + \frac{a_{10}(j)}{a_{10}(j)+a_{00}(j)}} \tag{2}$$

- The Ochiai coefficient:

$$s_O(j) = \frac{a_{11}(j)}{\sqrt{(a_{11}(j) + a_{01}(j)) * (a_{11}(j) + a_{10}(j))}} \tag{3}$$

where $a_{pq}(j) =\mid \{i \mid x_{ij} = p \wedge e_i = q\} \mid$, and $p, q \in 0, 1$. The columns with the highest similarity coefficients are considered as the localization of j-th program code block.

2.2 Diagnostic Accuracy

In general, the following evaluation metrics is used for diagnostic accuracy [7]. Let $d \in \{1, \ldots, P\}$ be the index of the block that we know to contain a fault. For all

$j \in \{1, \ldots, P\}$, let s_j denote the similarity coefficient calculated for block j. Then the ranking position is given by

$$\tau = \frac{\mid \{j \mid s_j > s_d\} \mid + \mid \{j \mid s_j \geq s_d\} \mid -1}{2} \tag{4}$$

Accuracy, the percentage of blocks that need not to be considered when searching for the fault by traversing the ranking, is defined as

$$q_d = \left(1 - \frac{\tau}{P-1}\right).100\% \tag{5}$$

3 Experiments

This section describes the toy experiment of ASR program spectra-based localization. The objective of the experiment was as follows: we injected an accuracy drop between several batch runs of the test under different conditions and the task was to accurately localize a part of the program responsible for the accuracy drop. A free speech recognizer[1] was used for logatome recognition with uniform distribution of grammar weights. Logatome recognition task was selected as a standard diagnostic task with the vocabulary consisting of CVC isolated words in comparable quantity as used by Steeneken in his diagnostic work [8], and we use it often for validation of our diagnostic methods.

3.1 ASR Setup

The speech database used for ASR experiments is the Oldenburg Logatome Corpus (OLLO) [9]. It contains 150 different non-sense utterances (logatomes) spoken by 40 German and 10 French speakers. Each logatome consists of the combination of consonant-vowel-consonant (CVC) or vowel-consonant-vowel (VCV) with the outer phonemes being identical. To provide an insight into the influence of speech intrinsic variabilities on speech recognition, OLLO covers several variabilities such as speaking rate and effort, dialect, accent and speaking style (statement and question). The OLLO database is freely available at `http://sirius.physik.uni-oldenburg.de`.

Each of the 150 logatomes was recorded in six conditions (speaking rate: fast and slow; speaking effort: loud and soft; speaking style: spoken as question and normal) with three repetitions. This results in 2,700 logatomes per speaker. Influences of dialect may be investigated, as speakers without a clearly marked dialect and from four different dialect/accent regions were recorded. Utterances of ten speakers with no accented speech (five speakers for training and five speakers for testing) were used for ASR tests.

A Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM) based speech recognition system was trained using public domain machine-learning library TORCH [10] on the training set that consists of 13,446 logatome utterances. Three

[1] The source code is freely accessible at `http://www.torch.ch`

states left-right HMM models were trained for each of the 26 phonemes in the OLLO database including silence as well. Three acoustic models (AMs) were trained, with 8, 16 and 32 GMMs, respectively. Diagonal covariance matrices were used to model the emission probability densities of the 39 dimensional feature vectors – 13 cepstral coefficients and their derivatives (Δs) and double derivatives ($\Delta\Delta s$). The phoneme HMMs were connected with no skip. We trained and tested the ASR system with MFCC feature set, calculated using HTK `hcopy` tool. We calculated MFCC vectors every 10 msec using windows of size 25 msec.

3.2 Error Injection in Grammar Constrains

The test set size was 13.5k logatomes. We injected accuracy drop with (de)activation of isolated word recognition within an experiment: running isolated logatome recognition the system's accuracy was higher than running the test with the word loop grammar. To test the robustness of the localization, 3 experiments were performed, each with different AM. Performances of the tests are shown in Table 1.

Table 1. Accuracies of particular experiments

Experiments	AM	isolated	non-isolated
Experiment 1	8 GMM	71.83	69.40
Experiment 2	16 GMM	74.96	72.94
Experiment 3	32 GMM	75.87	73.39

The task was to construct program spectra for each experiment from R runs (13.5k) of isolated and R runs of non-isolated logatome recognitions (this constituted $2R \times P$ program spectra matrices), and to localize a part of the ASR implementation responsible for the accuracy drop. Figure 2 lists a simplified grammar construction used in the experiments. Program lines 6–8 constitute the part of the program executed for non-isolated word recognition grammar.

A program spectrum was obtained using the standard GNU code coverage analysis tool `gcov`. This tool was used in conjunction with `gcc` to dump run-time code coverage of the ASR. The ASR was compiled with the `gcc -fprofile-arcs -ftest-coverage` options and no optimizations, and with each execution of the program, a separate data file `.gcda` was created for each object file. Using data files and source files `gcov` tool then generated profiling `.gcov` files, which described which lines of code were actually executed and how often each line of code was executed. This information was used for program spectra matrix construction, where lines of code were transformed to columns with a binary value indicating the execution or non execution of the line in a particular program run. Program executions formed rows of the matrix. The last column of the matrix represented recognition failures (detected insertions, deletions or substitutions) with values of 1 for errors detected.

We generated three spectra, each for different experiment. The first part of the spectra was from isolated logatome recognition, and the second part of the spectra was from

```
// from leading silence to a start state
gramm   ar.transitions[1][0] = true;
for (int i=0;i<n_words-1;i++) {
 // from the word to the leading silence
 grammar.transitions[i+2][1] = true;
 // from the trailing silence to the word
 grammar.transitions[n_words+1][i+2] = true;
 if (!isolated) {
  /* A block of interest */
  for (int j=0;j<n_words-1;j++)
   if (!no_self_transitions || i!=j)
    // word to word transition
    grammar.transitions[j+2][i+2] = true;
 }
}
grammar.transitions[n_words+2][n_words+1] = true;
```

Fig. 2. Simple grammar construction code

non-isolated recognition. We calculated $s_J(j)$, $s_T(j)$, and $s_O(j)$ similarity coefficients for the obtained spectra. We chose the parameter of different number of mixtures in GMM models just for getting more tests with different WER. Because we ran speech recognition program R times for each test, we also obtained static parts of the spectra ($x_{ij} = 1$, where $\forall i$ is j constant $j = k$, $k \in \{1, \ldots, P\}$). Such a static spectra must

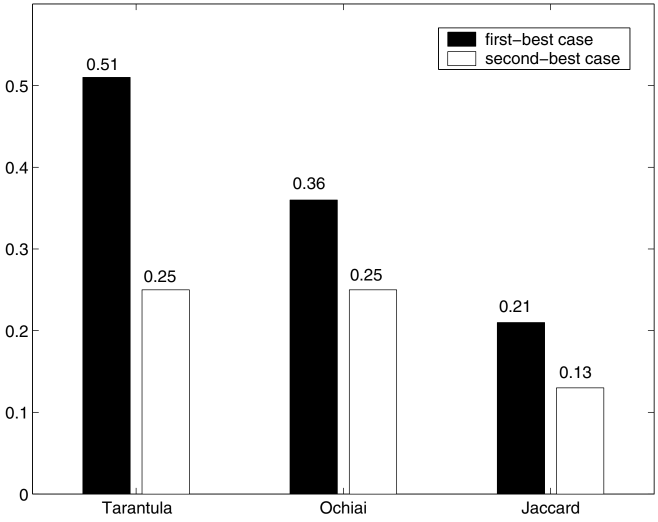


Fig. 3. Differences in scoring first two localization candidates for 16GMM test case

be penalized to achieve good diagnostic performance, and we estimated this penalty as $s_j = \frac{1}{n} * s_j$, where n is a number of ASR parameter sets in the experiment. In our case $n = 2$, as we first ran non-word isolated speech recognition, and then word isolated speech recognition.

We achieved good diagnostic accuracy q_{6-8} in our experiments (the index of q represents the code lines from Fig. 2). The accuracy was 99.995% for all the similarity measures and for the tests, except for $s_J(j)$ and $s_O(j)$ in 8GMM test case, where q_{6-8} was 75.51%.

In addition to q_{6-8}, Fig. 3 shows how the measures scored first-best and second-best cases for fault localization. Tarantula measure $s_T(j)$, the only measure with $q_{6-8} = 99.995\%$ for all three test cases (AMs), has also the highest difference in scoring of first two localization candidates. This difference may also indicate the quality of fault localization.

4 Discussion

The presented fine-grained localization has the following advantages: First, it can be applied without any code modification in a fully automatic manner, as it is a black-box diagnosis method applicable to any ASR system without providing access to individual components within the system. Second, the diagnosis is easily implemented using existing profiling tools (such as Linux `gcov` or Windows `VSPerfMon`). Third, if the program spectrum and its further processing is well designed, the method offers the finest localization revealing possible errors ranging from software bugs via badly tuned ASR parameters toward the ultimate goal of providing an insight into erroneous behavior of computer speech recognition.

However, there are still limitations to the application of the diagnosis as described in this paper. Faults, that does not change the program spectra with a failure, cannot be detected. As it sticks to the implementation, even the best code in the world will fail with poor acoustical and language models used; but this can be well detected by existing coarse-grained decision-tree-based diagnosis methods. It seems that their combination for complex diagnosis is needed, and we plan to examine this in the future.

The diagnosis is open to using different runtime coverage of entities such as statements, branches and du-pair coverage. Experiments show that the choice of coverage type can significantly affect the effectiveness of fault localization [11]. In addition, different similarity measures can be further proposed. It would be worth to predict more localities that all simultaneously contribute to the error. For now only single locality was considered. All of these issues show the directions of our future work.

In the spirit of making research reproducible [12], the diagnostic tools used, including batch scripts, data descriptions, ASR and diagnosis setup, are available at `http://www.ui.savba.sk/speech/milos_web_data/asr_diagnosis.zip`.

Acknowledgements. The work has been funded with support from the VEGA grant No. 2/0138/08 entitled "Robust speech technologies for information systems in Slovak and their diagnostics", and by Slovak Research and Development Agency under research project APVV-0369-07.

References

1. Chase L. L.: Error-Responsive Feedback Mechanisms for Speech Recognizers. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh (April 1997); Also available as Robotics Institute Tech. Report # CMU-RI-TR-97-18
2. Nanjo, H., Lee, A., Kawahara, T.: Automatic Diagnosis of Recognition Errors in LVCSR Systems. In: ICSLP 2000, vol. 2, pp. 1027–1030 (2000)
3. Cerňak, M.: A Comparison of Decision Tree Classifiers for Automatic Diagnosis of Speech Recognition Errors. Computing and Informatics 3 (July 2010)
4. Zheng, A.X., Lloyd, J., Brewer, E.: Failure Diagnosis using Decision Trees. In: Proceedings of the First International Conference on Autonomic Computing (ICAC 2004), Washington, DC, pp. 36–43. IEEE Computer Society Press, Los Alamitos (2004)
5. Picheny, M., Nahamoo, D.: Towards Superhuman Speech Recognition. In: Benesty, J., Sondhi, M.M., Huang, Y. (eds.) Springer Handbook of Speech Proccesing, pp. 597–616. Springer, Heidelberg (2008)
6. Zoeteweij, P., Abreu, R., van Gemund, A.J.C.: Software Fault Diagnosis. In: A Tutorial in TESTCOM/FATES, Tallinn, Estonia, June 26-29 (2007)
7. Abreu, R., Zoeteweij, P., van Gemund, A.J.C.: On the Accuracy of Spectrum-based Fault Localization. In: Proceedings of TAIC PART (2007)
8. Steeneken, H.J.M., Varga, A.: Assessment for Automatic Speech Recognition. Comparison of Assessment Methods, Speech Communication 12(3), 241–246 (1993)
9. Wesker, T., Meyer, B., Wagener, J., Anemuller, J., Mertins, A., Kollmeier, B.: Oldenburg Logatome Speech Corpus (OLLO) for Speech Recognition Experiments with Humans and Machines. In: Interspeech, pp. 1273–1276 (2005)
10. Collobert, R., Bengio, S., Mariéthoz, J.: Torch: a Modular Machine Learning Software Library. Technical Report IDIAP-RR 02-46, IDIAP (2002)
11. Santelices, R., Jones, J.A., Yu, Y., Harrold, M.J.: Lightweight Fault-Localization Using Multiple Coverage Types. In: Proc. of the ICSE, pp. 56–66 (2009)
12. Vandewalle, P., Kovačević, J., Vetterli, M.: Reproducible Research in Signal Processing: What, Why, and How. IEEE Signal Processing Magazine (37) (May 2009)

Automatic Lip Reading in the Dutch Language Using Active Appearance Models on High Speed Recordings

Alin Gavril Chitu, Karin Driel, and Leon J.M. Rothkrantz

Delft University of Technology,
Man-Machine Interaction Group, Department Mediamatica
Mekelweg 4, 2628CD, Delft, The Netherlands
{a.g.chitu,k.driel,l.j.m.rothkrantz}@tudelft.nl
http://mmi.tudelft.nl

Abstract. This paper presents our work on lip reading in the Dutch language. The results are based on a new data corpus recorded at 100Hz in our group. The NDUTAVSC corpus is to date the largest corpus build for lip reading in Dutch. For parameterising the input data we use Active Appearance Models. Based on the results of AAM we define a set of high level geometric features which are used for training recognizer systems for different recognition tasks, such as fixed length digits strings, random length letters strings, random word sequences, fixed topic continuous speech and random continuous speech. We show that our approach gives great improvements compared to previous results. We also investigate the influence of the high speed recordings on the performance of the recognition. We show that in the case of high speech rate the use of higher speed recordings is compulsory.

Keywords: lip reading, active appearance models, high speed recordings, data corpus, NDUTAVSC, Dutch.

1 Introduction

Today's trend is to make the interaction between humans and their artificial assistants easier and closer to the natural means of human communication. Speech recognition technology has reached a maximum of performance and good recipes for building speech recognizes have been written. However, the two major problems of background noise and reverberations are still insurmountable. Therefore, inspecting other sources for complementary information which could diminish these problems is a thinkable solution. Lip reading can therefore be seen both as a complementary process to speech recognition, and as a stand-alone application. The applications for lip reading are diverse: multi-media phones for the hearing impaired, mobile phone interface for public spaces, person identification, recovery of speech from deteriorated or mute movie clips, security by "video surveillance", etc.

In this paper we introduce our work towards automatic and robust lip reading. Data acquisition is presented in Section 4. This section describes the corpus specially developed for the current research. Data parametrisation is covered in Sections 5 and 6. The experiments and the results are given in Section 7. The future developments are discussed in Section 8. Section 3 describes the methodology used for building and testing a lip reader. The next section gives a succinct overview of the related work.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 259–266, 2010.

2 Related Work

Lip reading literature has increased rapidly over the years. There are two possible research directions, namely data parametrisation and inference mechanisms. Due to the temporal dynamics of speech, the majority of techniques used in other machine learning domains, template matching, rule based methods, decision trees, support vector machines, etc. are not suitable. Only in some limited vocabulary applications they could still be feasible. For instance, Wang et al. in [1] used b-spline functions to match the spoken utterances. For large vocabulary (continuous) speech recognition, however, a time series approach should be used. Due to its success in speech recognition the most used method is the Hidden Markov Models method and its derivatives. For lip feature extraction many methods have been developed. They fit mainly in three broad classes: texture methods, geometrical methods and combinations of them. The appearance-based methods consider a transformation of the raw image data as features for the recognition. The transformation of the raw image is employed in order to obtain some data reduction. The most popular method for this is Principal Component Analysis (PCA) [2]. As an alternative to PCA, a discrete cosine transform [3] or discrete wavelet transform have been applied. However, these approaches give rise to a very high dimensionality of the feature vectors. On the other side geometrical methods aim to model the visual parts of the speech production apparatus, such as lips, teeth, and tongue, but also other parts of the face. Usually, specific points on the face are detected and tracked. The detection process is often assisted by 2D or 3D geometrical models of the face [4]. Alternatively, statistical methods based on image filtering algorithms can be used to directly extract the shape of the lips. The dimensionality reduction obtained through this approach is very large. Performing optical flow analysis on the input video is used both as a measure of the overall movement (e.g for onset/offset detection [5]) or as feature extraction method [6].

3 Methodology

Building and a lip reader is in many respects similar to that of any pattern recognition system. There are four important stages: data acquisition, lip tracking, visual feature definition and training. A visual overview of the recognition process is shown in Figure 1. Data acquisition consists of gathering of a large corpus containing recordings of people speaking. The same as the speech recognition systems a lip reader is a data driven system which means that we need tens of thousand of video recordings in order to obtain good training. The next step is data parametrisation which reduces the dimensionality of the problem and captures the relevant information about the spoken utterance. Based on the visual features collected from every frame of every video the inference models are trained using a supervised learning approach. For inference we used the Hidden Markov Model (HMM) approach. Each viseme was modelled by a Left-Right HMM with five states, but only three emitting states in the middle. Each observation node was modelled with a set of up to 32 Gaussian mixtures. Intra word tri-visemes were computed in order to build context for the models. For testing the process differs only in the last stage when the trained models are used to recognize the spoken utterance based on the feature vectors computed from the input video.

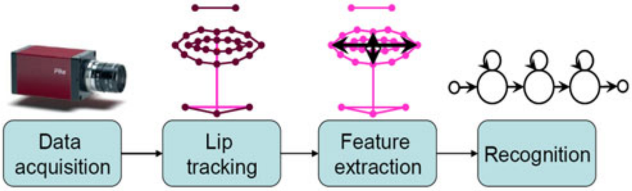


Fig. 1. Overview of the lip reading process

4 Data Acquisition

The data corpus is the foundation of any successful speech research. After working for some time with a small data corpus we arrived at the conclusion that a new, larger and more complex data corpus was needed. We did extensive analyses to understand what the drawbacks of the existing corpora were. The paper [7] provides a rich comparison among some of the most used data corpora and introduces a set of guidelines to be followed when building a data corpus, while paper [8] shows that in the case of high speech rate the influence of the video frame rate is extremely high. It was computed that when the video recording is performed at 25Hz in the case of high speech rate there are in average only three frames per viseme. Therefore, for the current research we build a new corpus which should overcome most of the problems we encountered with the old corpus. As shown in Figure 2, in the new corpus, recorded at 100Hz, in the case of high speech rate (i.e. more then 160 words per minute) the mean number of frames per viseme is around 8. For English the only reference known to us to a corpus with up to 60Hz is [12]. However, this is the first high speed corpus for Dutch.

4.1 Recording Settings

We considered recording in a controlled environment, namely having reasonable noise levels and good illumination. We recorded, therefore, synchronized dual-view of the lower half of the speaker's face at 100Hz and half PAL resolution. The recordings were controlled using a prompter which controlled the devices and instructed the speaker about the speaking style required. A sketch of the recording settings is shown in Figure 3.

4.2 Corpus Statistics

The new data corpus NDUTAVSC(The New Delft University of Technology Audio Visual Speech Corpus) is the largest corpus for lip reading in Dutch. It contains a large pool of words and phonetically rich sentences. For each session, the speaker was asked to utter different items (random sentences(RS), random digit strings(CD), random letters strings(CL), spellings, open questions(RS)) divided into categories with respect to the language content and speech style: normal rate, fast rate or whispering. The resulted corpus consists of 10 hours and 38 minutes of continuous recordings of 66 speakers, 20 females and 46 males.

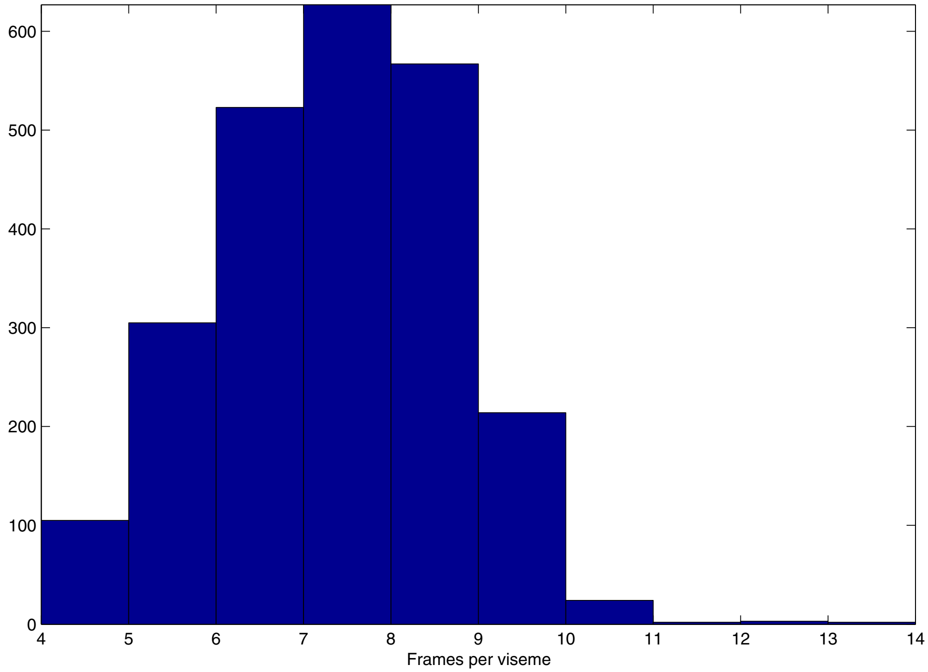


Fig. 2. Viseme coverage in the corpus NDUTAVSC. The histogram was computed using only the high speech utterances.

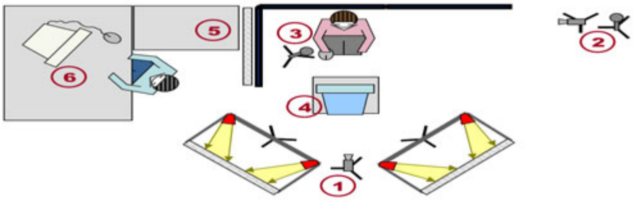


Fig. 3. The sketch of the recordings setup: 1) frontal camera (height 1.20m); 2) side camera (height 1.20m) and environment microphone (height 1.50m); 3) speaker and the main microphone (height 80cm); 4) monitor showing the prompts; 5) computer controlling the recordings (devices and utterances); 6) operator supervising the recordings. The two panels were used to cast a diffuse light on the speaker's face.

5 Active Appearance Models

Active Appearance Models (AAM) is a model based approach for image segmentation. It was first introduced by Edwards, Taylor and Cootes in [10] for interpreting face images. It is a top-down approach because it is based on a-priori knowledge about shape and appearance of the targeted object, therefore, it can only be applied in situations when the test images are already validated to contain the targeted object. During the training phase a statistical model of the shape and appearance of the object is built starting from the variation induced by the physical phenomena that governs the object's image. The powerful generalization to unseen instances makes them extremely useful in various applications, however, the most important being in the medical domain. A good overview on the AAM is done in [11]. The method assumes that a set of "landmark" points can always be marked on the image to describe the shape of the object. The transformation of the object should, therefore, not degenerate such that the points overlap or are occluded. From a set of images which have the landmarks already marked the method generates a statistical model of the shape variation, a model of the texture variation and a model of the correlations between the shape and the texture. Based on this models the method should be able to synthesize any new instance of the target object, even though it was not "seen" in the training set, as long the variation from the mean shape of the object is in the learned dynamic range. The searching schema consists of an iterative model refinement that adjusts the model parameters such that the error between the synthesized model image and the real image is minimized. The AAM is based on PCA which is used three times during analysis: for shape analysis, for texture analysis and for the combination of shape and texture.

6 Features Extraction

The AAM model parameters can be used directly as feature vectors entries. We opted, however, to define based on the key points detected using AAM a set of high geometric features. This parametrisation has the advantage of providing the inference engine with data that encapsulates the most important aspects of speech. This also acts as a dimensionality reduction procedure since the dimension of the feature space is lower than the dimension of the image space. The AAM model used is shown in the left image in Figure 4. The image on the right shows the landmarks that define the model and are going to be used for computing the feature vectors. The visual features are defined as certain Euclidean distances and areas between the key points. These features are defined in Figure 5.

Figure 6 shows the plots of the feature vectors for four letters of the alphabet. We see that the variability of the features is very high which makes them suitable for the recognition task at hand. Even though the viseme "aa" is present in all first three letters we can clearly see that there is a slight difference between them with respect to the their

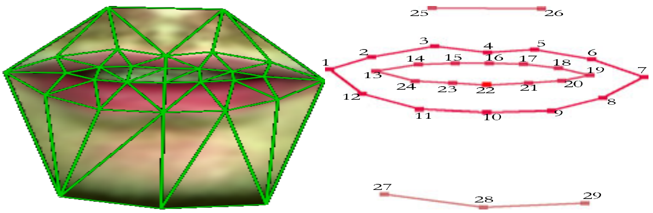


Fig. 4. The AMM used for feature extraction

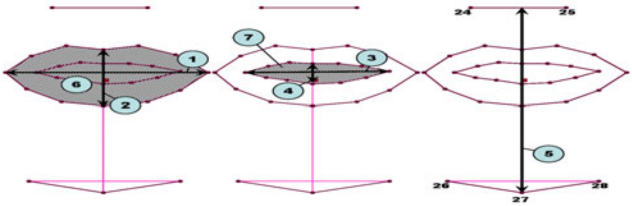


Fig. 5. Feature definition: 1) Outer lip width, 2) Outer lip height, 3) Inner lip width, 4) Inner lip height; 5) Chin to nose distance, 6) Outer lip area, 7) Inner lip area

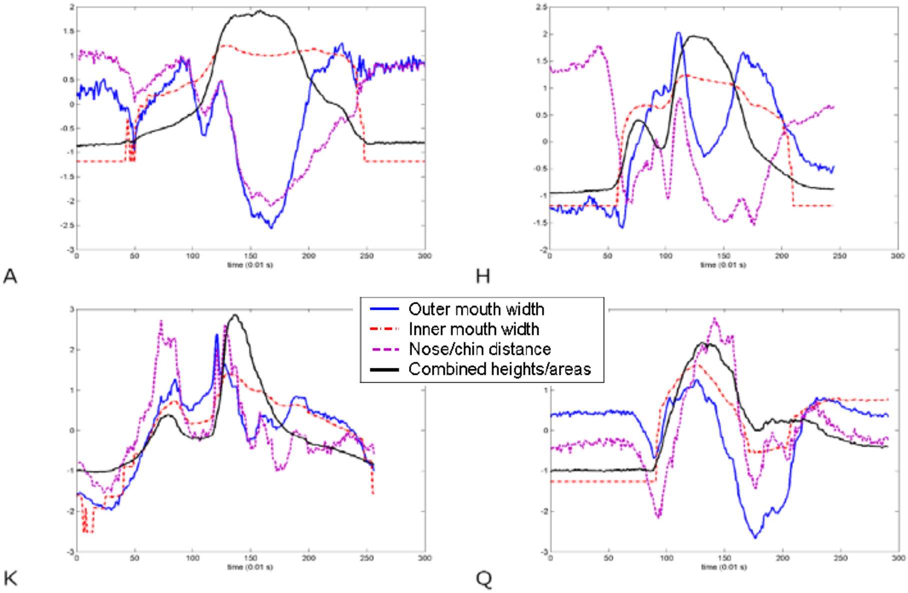


Fig. 6. Feature values plotted for the letters A (aa), H (h aa), K (gkx aa) and Q (gkx oyu). The vectors are scaled using the time variance and centred around their mean.

duration. This is best visible in the curve showing the height of the mouth, which shows that the duration of the viseme is shorter in the utterance of the letter “k” and “h” then in the case of the letter “a”.

An interesting result was obtained when visually inspecting the curves described by the feature vectors for all the visemes. By simple visual inspection we could easily distinguish between some of the visemes, which proved that the feature set captures much of the speech related information. Table 1 summarises our findings in this respect.

Table 1. Feature patterns per viseme: +) peak −) valley −+) increase +−) decrease

	aa	h	gkx	a	oyu	ie	ei	iee	td	sz	eeh	l	pbm	fvw	at
Outer mouth width	−		+	−	−	+	−+	+	+	+	+−	+−		+−	
Inner mouth width	+		+	+		+	+	+			+		−		
Nose/chin distance	−	+	+	−	−	−	−	+−	+		+−	+−			
Height/area features	+	+	+	+	−	+	+	+			+				

7 Experiments and Results

We divided the experiments based on the recognition tasks as defined in the data corpus. Two other tasks were added ’all’ which included everything and ’GU’ which included all utterances for which a fixed grammar could be written. The best results obtain on the 100Hz corpus are shown in table 2. We trained systems only on the static features but also on the static features together with dynamic features such as the first and second derivatives. As expected this boosts up considerably the recognition results. Compared to the digits strings in the case of letter strings the results are lower mainly because the number of classes is larger and, therefore, the problem more difficult to solve. However, a known issue with lip reading is that by the definition of the viseme the lip reading problem is inherently more difficult since the separability of the words in the visemes space is lower than in the phonemes space. This is because a viseme is defined as a set of phonemes and, therefore, a number of words end up having the same transcription in the viseme space. This is the case for six pairs of letters of the alphabet as transcribed in Dutch (P, B),(D,T), (G,J), (N,R), (O,U) and (V,W)). With respect to the recording frame rate we found out that the recognition performance peaks somewhere in the interval 25Hz to 50Hz and that in general 100Hz is too high. However, the peak of performance goes towards right as a function of the amount of high speech in the corpus. For instance in the case of CD task the peak is between 25Hz to 30Hz, for CL task the peak is between 30Hz to 35Hz while for GU task the peak is around 40Hz.

Table 2. Results of lip reading

Recognition Task	Static Features (7)	Extra Dynamic Features (7 + 14)
CD	69.86%	78.08%
CL	40.32%	49.46%
GU	33.39%	56.45%
RS	29.32%	38.52%
all	13.23%	25.08%

8 Future Work

Since our goal was to achieve continuous lip reading, we started new recording sessions which almost doubled the amount of data. We made plans as well to collect a large language corpus and construct a more reliable language model. Since we want to investigate as well the influence of the visual feature set on the performance of the lip reader, we use optical flow and other features extraction methods and compare the performance of the corresponding lip readers. Combining the frontal view with the side view in order to produce a 3D mouth model could further improve the performance. In a long run we envision a system that can process the recordings in real time, take advantage of the context in which the conversation is taken place and adapt accordingly.

References

1. Wang, S.L., Lau, W.H., Leung, S.H.: Automatic Lipreading with Limited Training Data. In: Proc. of the 18th Int. Conf. on Pattern Recognition, Washington, DC, vol. 3, pp. 881–884 (2006)
2. Bregler, C., Konig, Y.: “Eigenlip” for Robust Speech Recognition. In: IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (1994)
3. Duchnowski, P., Hunke, M., Büsching, D., Meier, U., Waibel, A.: Toward Movement-Invariant Automatic Lip-Reading and Speech Recognition. In: Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 1, pp. 109–112 (1995)
4. Essa, I.A., Pentland, A.: A Vision System for Observing and Extracting Facial Action Parameters. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 76–83 (1994)
5. Fleet, D.J., Black, M.J., Yacoob, Y., Jepson, A.D.: Design and Use of Linear Models for Image Motion Analysis. Int. Jour. of Computer Vision 36(3), 171–193 (2000)
6. Chiţu, A.G., Rothkrantz, L.J.M., Wiggers, P., Wojdel, J.: Comparison Between Different Feature Extraction Techniques for Audio-Visual. Jour. on Multimodal User Interfaces 1(1), 7–20 (2007)
7. Chiţu, A.G., Rothkrantz, L.J.M.: Building a Data Corpus for Audio-Visual Speech Recognition. In: Euromedia 2007, pp. 88–92 (2007)
8. Chiţu, A.G., Rothkrantz, L.J.M.: The Influence of Video Sampling Rate on Lipreading Performance. In: 12th Int. Conf. on Speech and Computer, pp. 678–684 (2007)
9. Wojdel, J.C., Wiggers, P., Rothkrantz, L.J.M.: An Audio-Visual Corpus for Multimodal Speech Recognition in Dutch Language. In: Proc. of the Int. Conf. on Spoken Language Processing, Denver CO, pp. 1917–1920 (2002)
10. Edwards, G., Taylor, C., Cootes, T.: Interpreting Face Images using Active Appearance Models. In: 3rd Int. Conf. on Automatic Face and Gesture Recognition, pp. 300–305 (1998)
11. Cootes, T.F., Taylor, C.J.: Statistical Models of Appearance for Medical Image Analysis and Computer Vision. In: Proc. of the SPIE Medical Imaging, vol. 4322, pp. 236–248 (2001)
12. Potamianos, G., Graf, H.P., Cosatto, E.: An Image Transform Approach for HMM Based Automatic Lipreading. In: Proc. of IEEE Int. Conf. on Image Processing, vol. 1, pp. 173–177 (1998)

Towards the Optimal Minimization of a Pronunciation Dictionary Model

Simon Dobrišek[1], Janez Žibert[2], and France Mihelič[1]

[1] University of Ljubljana, Faculty of Electrical Engineering,
Tržaška 25, SI-1000 Ljubljana, Slovenia
{simon.dobrisek,france.mihelic}@fe.uni-lj.si
[2] University of Primorska, Primorska Institute of Natural Sciences and Technology,
Muzejski trg 2, SI-6000 Koper, Slovenia
janez.zibert@upr.si

Abstract. This paper presents the results of our efforts to obtain the minimum possible finite-state representation of a pronunciation dictionary. Finite-state transducers are widely used to encode word pronunciations and our experiments revealed that the conventional redundancy-reduction algorithms developed within this framework yield suboptimal solutions. We found that the incremental construction and redundancy reduction of acyclic finite-state transducers creates considerably smaller models (up to 60%) than the conventional, non-incremental (batch) algorithms implemented in the OpenFST toolkit.

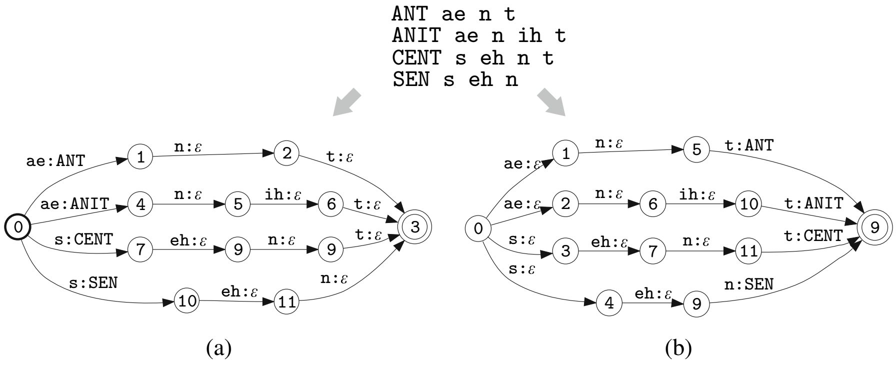
1 Introduction

A spoken-language model is commonly designed as a cascade of deterministic, non-deterministic and probabilistic finite-state transducers (FSTs). Such a composition can be seen as a unified, multi-level, probabilistic finite-state network, where different levels correspond to the various spoken-language constraints, such as grammar, lexicon, pronunciation rules, acoustic-phonetic models, etc [1]. This approach reduces the problem of automatic speech recognition (ASR) to the problem of searching for the most probable path through a finite-state network, given a sequence of acoustic speech observations [2].

Large-vocabulary (LV) ASR systems require very large finite-state models that may be composed of tens of millions of states and transitions. Any implementation of a LV-ASR system requires an optimization of the model in terms of its size and algorithmic complexity. The concept of weighted FSTs and the toolkits developed by Mohri et al. [3] provide a general representation and an algorithmic framework for such an optimization. The framework is implemented in the toolkit called OpenFST [4]. This toolkit provides efficient implementations of the conventional algorithms for constructing, combining, optimizing, and searching the generalized weighted FSTs.

We explored the concepts and algorithms provided by the OpenFST library, where we focused on the algorithms for optimizing FSTs in terms of size. We have limited our research to the construction of a pronunciation dictionary component of the whole finite-state model [5]. We experimented with unweighted acyclic FSTs to model large pronunciation dictionaries with hundreds of thousands of word forms. To our

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 267–274, 2010.



```
ANT ae n t
ANIT ae n ih t
CENT s eh n t
SEN s eh n
```

(a) (b)

Fig. 1. The two possible initial unweighted acyclic FST that were constructed from the given list of words with their pronunciation. The states are represented by circles and marked with their unique number. The initial state is represented by a bold circle and the final state by a double circle. The input label `i` and the output label `o` of a transition are marked on the corresponding directed arc by `i:o`.

surprise we found that the incremental construction and optimization of such FSTs yield considerably better results than the conventional non-incremental approach. We developed an alternative algorithm for the incremental construction and redundancy reduction of unweighted acyclic FSTs. The algorithm proved to be faster and creates considerably smaller FSTs than the original OpenFST algorithms.

2 Pronunciation Dictionary Model

Word pronunciations are assumed to be finite and consequently they can be encoded using acyclic FSTs. Without any loss of generality we restricted our experiments to the unweighted FSTs as the extension to the weighted FSTs seems to be straightforward.

Let us assume that the initial FST that represents the pronunciation dictionary is constructed from a list of words with their pronunciations, as depicted in Fig. 1. The symbol ε denotes the empty string that is defined as the identity element with the string concatenation operation. Accordingly, the output label that represents a word can be placed on any of the directed arcs between the initial and final state, and consequently, different FSTs can be constructed from the same word list. However, such different FSTs are considered to be equivalent since they encode the same word pronunciations.

Fig. 1 shows the two possible FSTs that are considered to be equivalent. We could say that the output labels of FST(a) are pushed towards the initial state, and the output labels of FST(b) are pushed towards the final state. Since we are restricted to the unweighted FSTs, there is no need to use different final states for each of the word pronunciations.

3 Redundancy Reduction

In a deterministic FST, no two transitions from the same state share the same input label, which is not the case with the two shown initial FSTs. A deterministic FST has

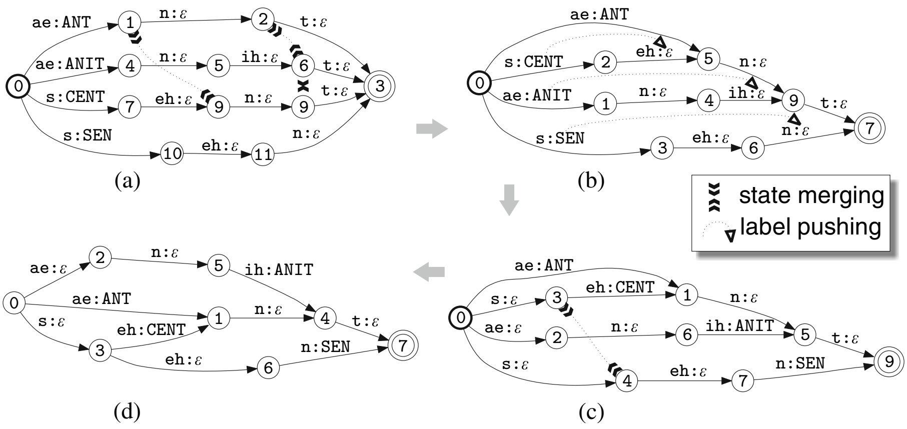


Fig. 2. The three steps of the FST redundancy reduction via the initial FSA minimisation, the intermediate output label pushing towards the final state, and the final FSA determinisation

an advantage over the equivalent nondeterministic one in terms of lower redundancy. As shown by Mohri et al. [1] the nondeterminism of a FST can be eliminated, or at least reduced, by first encoding it as a finite-state acceptor (FSA), i.e., each pair of input and output labels is treated as one label, and then using the classical FSA determinisation algorithm [6] the obtained nondeterministic FSA is transformed into an equivalent deterministic FSA.

The redundancy of the obtained deterministic FSA can be reduced even further by using the classical minimization algorithm that merges its equivalent states [7]. The two states of a deterministic FSA are said to be equivalent if exactly the same sequence of symbols labels the paths from these states to the final state. The final minimal FSA can then be decoded back as a FST.

However, this redundancy-reduction procedure via the FSA determinisation and minimisation does not yield the minimum possible FST being equivalent to the initial FST. The procedure has to be combined with the output label pushing that preserves the equivalency of the transformed FST and enables a further redundancy reduction. The combined procedure is illustrated in Fig. 2.

Fig. 2 shows the initial FST(a) constructed from the word pronunciation list given in Fig.1. The initial FST is encoded as a FSA, which is then determinised and minimised as described above. The determinisation step is actually not required since, due to the left-most position of the output labels in the initial FST(a), the initial FSA is already deterministic. Fig. 2 shows the FST(b) that resulted from the FSA minimisation. As can be seen from the figure, further redundancy reduction is possible if the output labels are pushed towards the final states as far as the equivalency of the transformed FST is still preserved. FST(c) is equivalent to FST(b), and since it is nondeterministic its redundancy can be further reduced. The reduction is performed by, again, first encoding the FST as a FSA, performing its determinisation and then obtaining the final FST(d) that is decoded back from the determinised FSA. The described procedure is conventional and can be performed using the algorithms that are implemented in the OpenFST toolkit.

An important observation that can be drawn from the analysis of the described procedure is that the final FST is not necessarily the minimum possible FST that is equivalent to the initial FST. The procedure consists of three steps that are optimized locally; however, they are mutually dependent. For instance, the suboptimal redundancy-reduction at the first minimisation step may lead to a better total redundancy reduction obtained at the third determinisation step. One could also start the procedure with the initial FST(b) shown in Fig. 1 and perform the FSA determinisation at the first step, the output label pushing towards the initial state at the second step, and the FSA minimisation at the last step. Our experiments revealed that the two possible resulting FSTs can have different sizes, even though they represent the same pronunciation dictionary.

4 Incremental FST Construction and Redundancy Reduction

The FST redundancy-reduction procedure discussed in the previous section is based on the size optimisation of the whole initial FSTs. An alternative approach would be to construct a FST and reduce its redundancy incrementally, word by word. In this procedure, a left-to-right single-path FST is created for each given word pronunciation. A union between this FST and the incrementally growing FST, which is being constructed to represent the complete pronunciation dictionary, is then created. After each such union, the described redundancy reduction can be performed as usual. This procedure is illustrated in Fig. 3.

The incremental procedure can be performed by combining the conventional algorithms implemented in the OpenFST toolkit; however, such a procedure has a very

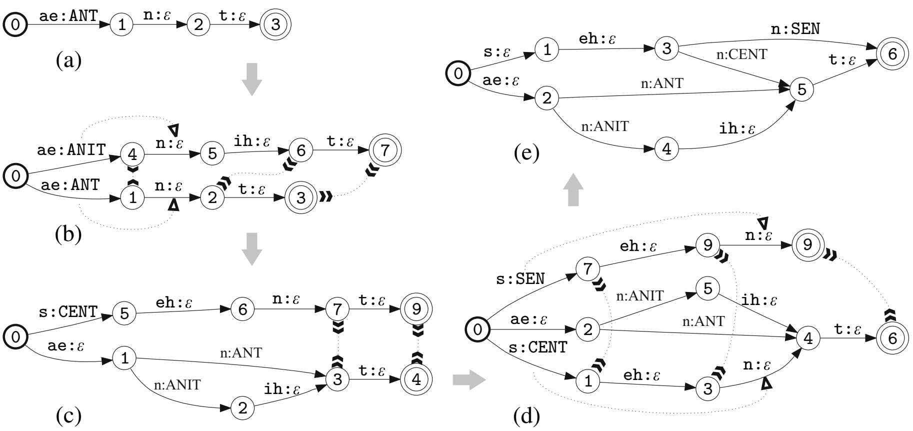


Fig. 3. Incremental construction of a FST from the list of word pronunciations. The usual redundancy reduction is performed after each union between the newly-added FST, representing the newly added word pronunciation, and the existing incrementally-growing FST being constructed to represent the complete pronunciation dictionary.

high (exponential-time) complexity. Our experiments showed that it can take several hours to create a FST from a 50k-word pronunciation dictionary. But to our surprise, the incremental procedure yields considerably smaller FSTs. This can be seen from the comparison of the given figures. The final FST(e) shown in Fig. 3 has one state and one transition less than the final FST(d) shown in Fig. 2; however, both FSTs represent the same pronunciation dictionary.

We developed an alternative algorithm for the incremental construction and redundancy reduction of acyclic FSTs that has a much lower (linear-time) complexity. The algorithm was actually developed independently from the algorithms that are provided in the OpenFST toolkit. For historical reasons (the existing computer code, etc) the first version of the proposed algorithm was developed for the Moore type of FSTs [5]. The algorithm is now improved to support the Mealy type of FSTs that are also supported by default by the OpenFST toolkit.

The algorithm incrementally constructs a FST by merging the newly added single-path FST, which represents the newly added word pronunciation, with the incrementally growing FST being constructed. The algorithm is composed of two parts that we call the incremental tail-merge minimisation and the incremental head-merge determinisation. The two algorithms are performed for each of the added word pronunciations.

First, a single-path FST that represents the newly added word pronunciation is constructed as usual. The output (word) label is put on the initial state transition. A union between the newly added FST and the incrementally growing FST is then performed, where the final states are merged into a single final state. The following two algorithms are then performed to reduce the FST redundancy.

The tail-merge minimisation: The algorithm iterates backwards from the final state to the newly added states and merges them with the existing states, if certain conditions are satisfied. The algorithm involves the following steps:

Step 1: The current state becomes the final state of the unified FST.

Step 2: For each of the current state's predecessors the following conditions are verified:

- The given predecessor state is not the initial state and also not the newly added predecessor state;
- The input label of the transition from the given predecessor state to the current state is the same as the input label of the transition from the newly added predecessor state to the current state;
- The given predecessor state has only one successor (the current state) and the transition to the current state has the ε-output label.

If all the above conditions are satisfied for the given predecessor state then:

- The given predecessor state and the newly added predecessor state are merged into one state, and all the transitions to and from the merged state are rearranged, as necessary;
- The current state becomes the merged state.
- Recurse back to **Step 2**.

Step 3: Quit and return the current state.

The head-merge determinisation: The algorithm is performed after the tail-merge minimisation. This algorithm performs state merging from the opposite side of the unified FST. It also performs output-label pushing from transitions to transitions, if necessary. The algorithm is defined as follows:

Step 1: The current state becomes the initial state of the unified FST.
Step 2: For each of the current state successors the following conditions are verified:
- The given successor state is not the last state that was merged by the tail-merge algorithm and also not the newly added successor state;
- The input label of the transitions to the given successor state is the same as the input label of the transitions to the newly added successor state;
- The transition to the given successor state has the ε-output label, or if not so, the given successor state has only one predecessor and also only one successor with the ε-output label transition.

If all the above conditions are satisfied for the given successor state then:
- If the transition to the given successor state or to the newly added successor state has a word-output label, then this output label is pushed forward to the successor's transition to its single successor;
- The given successor state and the newly added successor state are merged into one state, and all the transitions to and from the merged state are rearranged, as necessary;
- The current state becomes the merged state;
- Recurse back to **Step 2**.

Step 3: Quit and return the current state.

The presented algorithm constructs a FST in the same way as depicted in Fig. 3. It actually creates the same final FSTs as created by the generalized incremental procedure described at the beginning of this section.

In general, the presented incremental procedure also does not necessarily create the minimum possible FST that represents the given pronunciation dictionary. It is obvious that the final result depends on how the word sequence that is processed by the algorithms is ordered. We found experimentally that better results can be achieved if a given word pronunciation dictionary is not lexically ordered. Our experiments actually showed that the size of the obtained FSTs can be up to 10% smaller if the input word list is randomly shuffled.

5 Experimental Results

All the discussed algorithms were evaluated by performing experiments with several large pronunciation dictionaries. The OpenFST toolkit algorithms were used to perform the conventional non-incremental FST creation procedures and the obtained FST sizes were compared to the sizes obtained by the proposed alternative incremental algorithm.

In this paper, we report the results of experiments with dictionaries from three different language groups, namely, US English, Slovenian, and Italian. The first one is the well-known CMU Pronouncing Dictionary for North American English (CMU-US). The version we used contains over 133k phonetically transcribed words. The

Table 1. The sizes (number of states/number of transitions) of the FSTs obtained from the three pronunciation dictionaries

Optimisation	CMU-US (133k)	MTE-SL (201k)	FST-IT (401k)
None	717,512/850,868	1,688,273/1,890,225	3,903,866/4,314,710
OFST-SFI	80,648/214,004	106,947/308,899	352,739/763,489
OFST-SFF	64,002/197,358	115,141/317,093	237,253/647,779
ITHM-ORD	33,473/166,829	34,974/236,926	99,801/510,645
ITHM-SHF	33,206/166,562	19,689/221,641	38,511/449,355

second one is the Multext-East Slovene Dictionary (MTE-SL). The version we used contains over 201k words transcribed using our TTS grapheme-to-phoneme converter. The Italian pronunciation dictionary (FST-IT) was taken from the Festival TTS toolkit and it contains over 410k transcribed words.

Let OFST-SFI and OFST-SFF denote the two versions of the conventional non-incremental FST creation and the redundancy-reduction procedures that were performed using the OpenFST algorithms, as described in Sec. 3. The first version starts with the initial FST having output labels pushed towards the initial state and the second one starts with the initial FST having output labels pushed towards the final state.

Then let ITHM-ORD and ITHM-SHF denote the proposed incremental head- and tail-merging algorithms, where ITHM-ORD denotes the lexically ordered input pronunciation dictionary, and ITHM-SHF denotes the the input pronunciation dictionary with the randomly shuffled word list. Table 1 shows the sizes of the obtained FSTs that represent one of the three pronunciation dictionaries. As can be seen from the results, the proposed incremental algorithms create considerably smaller FSTs, especially for the more inflected languages, like Slovenian and Italian. This is to be expected since both languages have abundant inflections and many word forms with the same suffixes and/or prefixes.

The proposed incremental algorithm is also considerably faster than the conventional non-incremental procedure. For instance, our Java implementation [8] of the proposed algorithm generated a FST from the 201k-word example in less than 6 seconds on a common PC (the OpenFST non-incremental procedure implemented in C++ required almost 25 seconds). The correctness of the implementation was systematically verified by the algorithm that generates all the possible input/output sequences from the given FST and these sequences were compared to the original pronunciation dictionaries.

6 Conclusions

The presented incremental FST construction and redundancy-reduction procedure does not create the minimum possible FST for the given pronunciation dictionaries, however, the global minimum cannot be far from the presented results since the number of FST transitions is close to the number of words in the input dictionaries (see Table 1).

References

1. Mohri, M., Pereira, F., Riley, M.: Weighted Finite-state Transducers in Speech Recognition. Computer Speech and Language 16, 69–88 (2002)
2. Jelinek, F.: Statistical Methods for Speech Recognition. The MIT Press, Cambridge (1998)
3. Mohri, M., Pereira, F.C.N., Riley, M.: Speech Recognition with Weighted Finite-State Transducers. In: Rabiner, L., Juang, F. (eds.) Handbook on Speech Processing and Speech Communication, Part E: Speech recognition. Springer, Heidelberg (2008)
4. Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., Mohri, M.: OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In: Holub, J., Žd'árek, J. (eds.) CIAA 2007. LNCS, vol. 4783, pp. 11–23. Springer, Heidelberg (2007)
5. Dobrišek, S., Vesnicer, B., Mihelič, F.: A Sequential Minimization Algorithm for Finite-State Pronounciation Lexicon Models. In: Proceedings of Interspeech 2009, International Speech Communication Association, Brighton, UK, pp. 720–723 (2009)
6. Aho, A.V., Sethi, R., Ullman, J.D.: Compilers, Principles, Techniques and Tools. Addison Wesley, Reading (1986)
7. Revuz, D.: Minimisation of Acyclic Deterministic Automata in Linear Time. Theoretical Computer Science 92, 181–189 (1992)
8. Spider – The Demonstration Implementation of the Algorithm for the Incremental Construction and Redundancy Reduction of Unweighted Acyclic FSTs, http://luks.fe.uni-lj.si/spider
9. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.4). Cambridge University Engineering Department, Cambridge (2006)

Multimodal Emotion Recognition Based on the Decoupling of Emotion and Speaker Information

Rok Gajšek, Vitomir Štruc, and France Mihelič

Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, SI-1000 Ljubljana, Slovenia
{rok.gajsek,vitomir.struc,france.mihelic}@fe.uni-lj.si
http://luks.fe.uni-lj.si/

Abstract. The standard features used in emotion recognition carry, besides the emotion related information, also cues about the speaker. This is expected, since the nature of emotionally colored speech is similar to the variations in the speech signal, caused by different speakers. Therefore, we present a gradient descent derived transformation for the decoupling of emotion and speaker information contained in the acoustic features. The Interspeech '09 Emotion Challenge feature set is used as the baseline for the audio part. A similar procedure is employed on the video signal, where the nuisance attribute projection (NAP) is used to derive the transformation matrix, which contains information about the emotional state of the speaker. Ultimately, different NAP transformation matrices are compared using canonical correlations. The audio and video sub-systems are combined at the matching score level using different fusion techniques. The presented system is assessed on the publicly available eNTERFACE '05 database where significant improvements in the recognition performance are observed when compared to the stat-of-the-art baseline.

Keywords: speech, video, acoustic features, emotion recognition, multimodal databases.

1 Introduction

The focus of the speech recognition and computer vision communities on emotion or affect related topics, has been increasing over the past years. Findings in the field of automatic emotion analysis can benefit other areas of interest in human computer interaction (HCI) such as automatic speech recognition (ASR) systems where performance drops significantly when the speech is emotionally colored, dialog managers, where a detection of a frustration in user's speech could redirect the call to a human operator, etc. Our previous work [1] leads us to believe that the phenomena of emotions in speech is by its nature similar to the way the speaker specific information is conveyed by the speech signal. Therefore, in this work we evaluate the possibility of extracting speaker specific information from the features, thus increasing the accuracy of the emotion recognition performance. This was achieved by estimating a linear transformation that mapped the original feature vector extracted from the audio signal,

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 275–282, 2010.

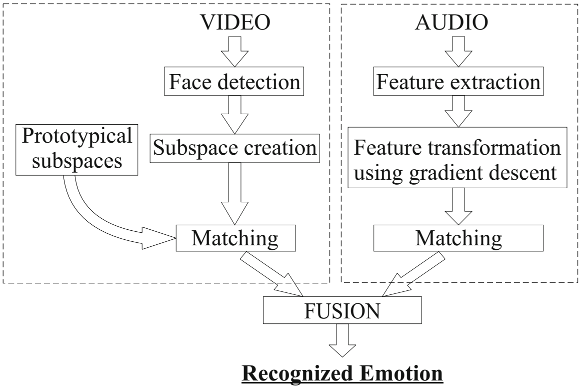


Fig. 1. Overvew of the multimodal emotion recognition system

to the mean vector of the appropriate speaker. The columns of the transformation matrix and the bias vector concatenated together formed a new feature vector.

A similar procedure was adopted for the video subsystem. Here, a subspace encoding the emotional state of the subject in the video sequence was constructed, and compared to the prototypical subspaces of the emotional classes in the database. The two subspaces were compared using an image-set based method relaying on the computation of canonical correlations.

The results from the audio subsystem were fused with the video subsystem in order to evaluate the overall increase in accuracy of our multimodal emotion recognition system. The eNTERFACE '05 multimodal database [2] was used to evaluate the final recognition performance.

2 Audio-Video Emotion Recognition System Architecture

The emotion recognition system consists of the two subsystems, one for each modality. Fig. 1 presents the structure of the system, where the left part represent the video subsystem and the right represents the audio part. Each systems performs the matching on its own and the scores for all classes are combined at the fusion level to produce the final score.

3 Audio Subsystem

The audio subsystem consists of three major parts, as presented in the right part of the Fig. 1. First, the low level acoustic features are extracted from the audio signal. Then, the procedure of extracting emotion related information from the features and discarding the identity information is performed. Finally, the matching algorithm produces the scores for each sample. The above steps are described in more detail in the following sections.

3.1 Acoustic Features

At the Interspeech '09 Emotion Challenge [3] the baseline feature set, consisting of 384 different features, was presented. In one part of the competition, where contestants were asked to produce their own feature sets that would surpass the baseline, there were none officially recognized contestants. This leads us to believe, that the proposed feature set forms the current stat-of-the-art in emotion recognition. In order to speed up the next step in the audio subsystem (Section 3.2) the whole feature set was reduced to 100 best features following the feature selection procedure based on mutual information as described in [4]. The comparison of the reduced and original feature set is presented in the Section 6.

3.2 Decoupling of Emotion and Speaker Specific Information

The idea of decoupling the emotional and speaker specific information is similar to the constrained version of MLLR (CMLLR) transformation [5] used for both speaker recognition and speech recognition (Eq. (1) shows the transformation of the mean vector in CMLLR).

$$\hat{\mu} = \mathbf{A}'\mu - \mathbf{b}' \tag{1}$$

Here, the matrix $\mathbf{A}$ and bias vector $\mathbf{b}$ are estimated for each speaker by increasing the likelihood of the acoustic model. This way, the speaker specific information is "moved" from the acoustic models representing base units (phones, triphones, etc.) to the transformation matrix and bias vector. Hence, the matrix $\mathbf{A}$ and vector $\mathbf{b}$ present an effective source of information for discriminating between different speakers. We propose a similar procedure of splitting the speaker specific information. The goal is to find a linear transformation which maps each speaker's sample to the mean value of all the samples from that particular speaker. Eq. (2) formulates the transformation, where $\hat{\mu}_i$ is the average of all the samples from the i-th speaker, $\mathbf{x}_{i,j}$ is the j-th sample from the speaker i, and matrix $\mathbf{A}_j$ and vector $\mathbf{b}_j$ represent the transformation for the sample j.

$$\hat{\mu}_i = \mathbf{A}_j\mathbf{x}_{i,j} + \mathbf{b}_j \tag{2}$$

Estimation of $\mathbf{A}$ and $\mathbf{b}$ is done employing a gradient descent based method. The cost function $J(\mathbf{A}_j, \mathbf{b}_j)$ for sample j is defined as shown in Eq. (3), where $\mathbf{x}_{i,j}$ is the j-th sample from the speaker i, and $\hat{\mu}_i$ is the average for speaker i.

$$J(\mathbf{A}_j, \mathbf{b}_j) = \sum_{d=1}^{D}(\mathbf{A}_j\mathbf{x}_{i,j}^d + \mathbf{b}_j - \mu_i)^2 \tag{3}$$

The partial derivatives of the cost function with respect to both variables $\mathbf{A}_j$ and $\mathbf{b}_j$ are shown in Eq. (5) and (4).

$$\frac{\partial J(\mathbf{A}_j, \mathbf{b}_j)}{\partial \mathbf{A}_j} = 2 * \sum_{d=1}^{D}(\mathbf{A}_j\mathbf{x}_{i,j}^d + \mathbf{b}_j - \hat{\mu}_i) * \mathbf{x}_{i,j} \tag{4}$$

$$\frac{\partial J(\mathbf{A}_j, \mathbf{b}_j)}{\partial \mathbf{b}_j} = 2 * \sum_{d=1}^{D} (\mathbf{A}_j \mathbf{x}_{i,j}^{d} + \mathbf{b}_j - \hat{\mu}_i) \tag{5}$$

In every iteration the new matrix $\hat{\mathbf{A}}_j$ and vector $\hat{\mathbf{b}}_j$ are estimated according to the Eq. (6) and (7).

$$\hat{\mathbf{A}}_j = \mathbf{A}_j - \delta_A \frac{\partial J(\mathbf{A}_j, \mathbf{b}_j)}{\partial \mathbf{A}_j} \tag{6}$$

$$\hat{\mathbf{b}}_j = \mathbf{b}_j - \delta_b \frac{\partial J(\mathbf{A}_j, \mathbf{b}_j)}{\partial \mathbf{b}_j} \tag{7}$$

When the cost function converges below the minimum threshold the final estimates of $\mathbf{A}_j$ and $\mathbf{b}_j$ are produced. Since the linear transformation converts each sample into the average feature vector for a specific features, the matrix $\mathbf{A}$ and vector $\mathbf{b}$ are believed to hold only the information about the speaker's emotional state. Therefore, the columns from the matrix $\mathbf{A}_j$ and the bias vector $\mathbf{b}_j$ are concatenated to form the new feature vector.

3.3 Matching of the Acoustic Features

For the task of producing the recognition scores, in order to enable the future fusion with video, support vector machines classifier (SVM) with a linear kernel, was employed. The one-versus-one approach was used for dealing with the five class recognition task. The setup used to produce the acoustic scores is described in more detail in Section 5.

4 Video Subsystem

The video subsystem comprises three main modules, as depict on the left hand side of Fig. 1: the face detection module, which detects and extracts the facial regions from the individual frames of the video sequence, the subspace creation module, which constructs a subspace from the given video sequence encoding mostly the emotion specific information, and finally, the matching module which compares the constructed subspace with the emotion-specific subspaces stored in the systems database. The described modules are presented in more detail in the remainder of this section.

4.1 Face Detection

Extraction and tracking of the facial region during the entire length of the given video sequence is done using the established Viola Jones face detector. The description of the detector would exceed the scope of this paper; however, the interested reader is referred to [6] for more information. An example of the output of the face detection module when applied on a sample video sequence is shown in Fig. 2.

4.2 Decoupling of Emotion and Speaker Information

The second module in the video subsystem represent the subspace creation module. Here, a subspace is created from the output images of the face detector in such a way

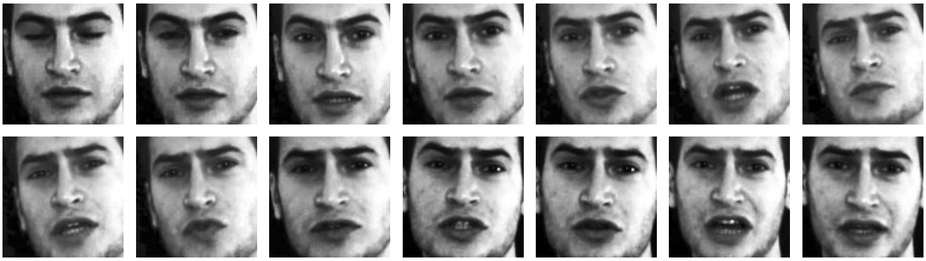


Fig. 2. An example of the output of the face detection module

that the emotion specific information in the subspace is enhanced, while the subject (or better said video sequence) specific information is decreased.

Let us assume that we have a set of facial images $\mathcal{X}_{\mathcal{Z}} = \{\mathbf{x}_i \in \mathbb{R}^d; \text{for } i = 1, 2, ..., n_{\mathcal{Z}}\}$ extracted from the given video sequence $\mathcal{Z}$. Here, $\mathbf{x}_i$ denotes the i-th d-dimensional facial image (in vector form) from the video sequence and $n_{\mathcal{Z}}$ stands for the number of frames in the sequence $\mathcal{Z}$. We assume that each of the $n_{\mathcal{Z}}$ facial images $\mathbf{x}_i$ can be decomposed into the following form:

$$\mathbf{x}_i = \hat{\mathbf{x}}_i + \mathbf{c}_i, \tag{8}$$

where $\hat{\mathbf{x}}_i$ represents the identity-specific (constant) part of the image $\mathbf{x}_i$, and $\mathbf{c}_i$ stands for the variable part of the image caused, for example, changes in the emotional state of the subject shown in the image.

Let us now assume that the variable part $\mathbf{c}_i$ of the image represents a random variable drawn from the normal distribution $\mathcal{N}(0, 1)$. It is possible to show that the video-sequence-conditional mean $\boldsymbol{\mu}_{\mathcal{Z}}$ represents an estimate of the constant identity-specific part of the images $\mathbf{x}_i$ (see [7] for details). Based on this observation we can conclude that if we remove the mean $\boldsymbol{\mu}_{\mathcal{Z}}$ from all facial images $\mathbf{x}_i$ comprising the set $\mathcal{X}_{\mathcal{Z}}$, we arrive at a new image set encoding only the variable (or channel/emotion) part of the video sequence, i.e.:

$$\mathcal{C}_{\mathcal{Z}} = \{\mathbf{c}_i = \mathbf{x}_i - \boldsymbol{\mu}_{\mathcal{Z}}; \text{for } i = 1, 2, ..., n_{\mathcal{Z}}\}. \tag{9}$$

To capture the variability of the channel images into a subspace that can be used for classification, we compute a scatter matrix $\mathbf{\Sigma}$ from the set of channel images $\mathcal{C}_{\mathcal{Z}}$. The first step here is the construction of the channel matrix $\mathbf{C} \in \mathbb{R}^{d \times n}$, i.e., $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n]$. This matrix is then employed for computation of the scatter matrix $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$:

$$\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T, \tag{10}$$

where T represents the transpose operator.

The subspace encoding the channel variations is finally determined by the leading eigenvectors (that correspond to non-zero eigenvalues) of the following eigenproblem:

$$\mathbf{\Sigma}\mathbf{w}_i = \lambda_i \mathbf{w}_i, \quad i = 1, 2, ..., d' \leq n. \tag{11}$$

If the scatter matrix is computed only from one test video sequence we obtain a subspace, that needs to be classified into one of the emotion classes. On the other hand, if the subspace is computed from all training video sequences of a given emotional state, we obtain the class prototypes (subspaces) for the specific emotion.

4.3 Matching the Subspaces

The last module in the video processing chain is the matching module, where the test subspace and prototype subspace are compared. Let us consider two d'-dimensional linear subspaces $\mathcal{W}_{\mathcal{Z}}$ and $\mathcal{W}_{\omega}$, where the subspace $\mathcal{W}_{\mathcal{Z}}$ can be thought of as a subspace extracted from a test video sequence and the subspace $\mathcal{W}_{\omega}$ represents the prototype subspace for the class labeled ω. We can measure the similarity of the two subspaces in terms of the canonical correlations, which are defined as cosines of principal angles $0 \leq \theta_1 \leq \theta_2 \leq ... \leq \theta_{d'} \leq (\pi/2)$, i.e.:

$$cos\theta_i = \max_{\mathbf{w}_{\mathcal{Z}i} \in \mathcal{W}_{\mathcal{Z}}} \max_{\mathbf{w}_{\omega i} \in \mathcal{W}_{\omega}} \mathbf{w}_{\mathcal{Z}i}^T \mathbf{w}_{\omega i}, \tag{12}$$

subject to $\mathbf{w}_{\mathcal{Z}i}^T \mathbf{w}_{\mathcal{Z}i} = \mathbf{w}_{\omega i}^T \mathbf{w}_{\omega i} = 1$, $\mathbf{w}_{\mathcal{Z}j}^T \mathbf{w}_{\mathcal{Z}i} = \mathbf{w}_{\omega j}^T \mathbf{w}_{\omega i} = 0$, for $i \neq j$ [8], where the vectors $\mathbf{w}_{\mathcal{Z}i}$ and $\mathbf{w}_{\omega i}$ denote the i-th basis vectors of the subspaces $\mathcal{W}_{\mathcal{Z}}$ and $\mathcal{W}_{\omega}$, respectively. The canonical correlations can be computed via Singular Value Decomposition (SVD) of the correlation matrix of the two subspaces. Let $\mathbf{W}_{\mathcal{Z}}$ and $\mathbf{W}_{\omega}$ stand for the matrices containing in their columns the orthonormal basis vectors of the subspaces $\mathcal{W}_{\mathcal{Z}}$ and $\mathcal{W}_{\omega}$. Then the SVD of the correlation matrix can be written as:

$$\mathbf{W}_{\mathcal{Z}}^T \mathbf{W}_{\omega} = \mathbf{Q}_{\mathcal{Z}\omega} \mathbf{\Lambda} \mathbf{Q}_{\omega\mathcal{Z}}, \tag{13}$$

where $\mathbf{\Lambda} = \text{diag}(cos\theta_1, cos\theta_2, ..., cos\theta_d')$ denotes the diagonal matrix of canonical correlations, and $\mathbf{Q}_{\mathcal{Z}\omega}$ and $\mathbf{Q}_{\omega\mathcal{Z}}$ represent orthogonal matrices.

The first canonical correlation accounts for the similarity of the closest two basis vectors of the two subspaces $\mathcal{W}_{\mathcal{Z}}$ and $\mathcal{W}_{\omega}$, while the remaining ones hold information about the proximity of the basis vectors in other dimensions [8], [9]. For classification purposes we use only the first (the maximum) canonical correlation and define the similarity between two subspaces as $\delta(\mathcal{W}_{\mathcal{Z}}, \mathcal{W}_{\omega}) = cos\theta_1$. Thus, we formulate the classification problem as follows:

$$\delta(\mathcal{W}_{\mathcal{Z}}, \mathcal{W}_{\omega_k}) = max_{i=1}^{N} \delta(\mathcal{W}_{\mathcal{Z}}, \mathcal{W}_{\omega_i}) \mapsto \mathcal{W}_{\mathcal{Z}} \in \omega_k. \tag{14}$$

The above expression postulates that if the similarity between the subspaces $\mathcal{W}_{\mathcal{Z}}$ and $\mathcal{W}_{\omega_k}$ is the highest among the similarities to all N subspaces then the subspace $\mathcal{W}_{\mathcal{Z}}$ is assigned to the k-th class.

5 Experimental Setup

The tests were conducted using the eNTERFACE '05 multimodal database [2], which consists of 6 different emotion classes. The five fold cross validation protocol was used, where in each fold 80% of samples were used for training, 10% comprised the development set for training the parameters of fusion, and 10% were used for testing.

The openSMILE toolkit [10] was used to produce the 384 features as described in Section 3.1. The feature set consists of spectral features (1–12 MFCCs), prosodic features

(F0, energy), voice quality features (harmonics to noise ratio) and zero-crossing-rate (ZRC). To this low level descriptors, 12 functionals are applied, thus producing the starting feature vector for each sample recording. Next, the number of features was reduced to only 100 most discriminative ones, using the algorithm described in [4]. This step was undertaken not to improve the classification scores, but to enable faster computation times for the estimation of linear transformations in the next step. As described in Section 3.2, the matrix **A** and vector **b** are estimated in order to transform each sample in to the mean feature vector for the corresponding speaker. Hence, the transformation contains more concentrated information about the emotional state, and the speaker specifics are discard. Since the dimension of a feature vector at this step is of size 100, the transformation matrix **A** is of size 100 × 100, and bias vector **b** is of size 100. Concatenating the columns of **A** and the vector **b**, thus comprise a new feature vector of size 10,100. For the gradient descent procedure, an identity matrix of size 100 × 100 was used for the initial value of **A**, and a vector of all ones for the starting value of **b**.

In the work, presented in this article we did not use a speaker identification system, which would first predict who the speaker in the sample is in order to select the appropriate speaker's mean vector. Instead, we presumed that we know the identity of the speaker and automatically selected the corresponding speaker's mean vector. The described approach was selected in order to evaluate the assumption of speaker specific information being correlated with the emotional state.

A version of sequential minimal optimization SVM was used to produce a model for each pair of emotion classes, following a one-versus-one protocol. Counting the number of wins for each emotion, the score for each sample is produced. Both, the video and the audio scores were normalized using min-max normalization [11]. A product rule fusion was used to combine the matching scores from both subsystems. The parameters of fusion were first determined on the development set, and the final recognition scores were produced on the test set.

6 Results

The standard measure of accuracy in emotion recognition systems has become the unweighted average recall, since it is useful in systems where the emotional classes are not balanced, which is usually the case in databases of spontaneous emotions. In our case the database is balanced, but in order to make our results comparable to the others in the literature, all the results presented in the Table 1 are unweighted average recalls over all emotion classes.

First, we evaluated the recognition performance of the original acoustic feature set where a recognition accuracy of 61% was achieved. With the reduction of the number of features from 384 to only 100 most discriminative ones, the recognition

Table 1. Comparison of average recalls for audio, video and multimodal emotion recognition

AUDIO subsystem			VIDEO	FUSION
original features (384)	reduced features (100)	decoupled features	subsystem	
61.2%	57.01%	**66.03%**	**54.61%**	**74.33%**

rate deteriorated, and the average recall over all folds dropped for approximately 4% absolute. But with the proposed decoupling of speaker specific information using gradient descent method the recognition accuracy jumped to 66.03%, which is a relative increase of 15%. After the fusion with the scores from the video subsystem, which achieves on its own an accuracy of 54.61%, the final recognition performance of our system climbs to 74.33%.

7 Conclusion

In the paper we presented a method of decoupling the emotion and speaker specific information form the acoustic features, usually used in an emotion recognition systems. The proposed method was evaluated on a popular multimodal database eNTERFACE'05, which enabled the fusion of audio and video. We have shown that if we use a smaller set of features (reducing from 384 to 100) in combination with the proposed method of extracting emotion specific information and discard the speaker information, we can achieve an increase in recognition performance of 15% over the reduced feature set. Over the baseline system we increased the recognition performance by 8% relative. In the future we will focus on modifications of the proposed algorithm in order to be able to use it on a larger set of baseline features.

References

1. Gajšek, R., Štruc, V., Dobrišek, S., Mihelič, F.: Emotion Recognition Using Linear Transformations in Combination with Video. In: Proceedings of Interspeech 2009 (2009)
2. Martin, O., Kotsia, I., Macq, B., Pitas, I.: The Enterface 2005 Audio-Visual Emotion Database. In: ICDEW 2006: Proceedings of the 22nd International Conference on Data Engineering Workshops, Washington, DC, USA. IEEE Computer Society Press, Los Alamitos (2006)
3. Schuller, B., Steidl, S., Batliner, A.: The Interspeech 2009 Emotion Challenge. In: ISCA (ed.), Proceedings of Interspeech 2009, pp. 312–315 (2009)
4. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226–1238 (2005)
5. Gales, M.J.F.: Maximum likelihood linear transformations for HMM-based speech recognition. Computer Speech and Language 12, 75–98 (1997)
6. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
7. Štruc, V., Vesnicer, B., Mihelič, F., Pavešić, N.: Removing illumination artefact from face images using the nuisance attribute projection. In: ICASSP 2010, pp. 846–849 (2010)
8. Kim, T., Kittler, J., Cipolla, R.: Discriminative learning and recognition of image set classes using canonical correlations. TPAMI 29, 1005–1018 (2007)
9. Yamaguchi, O., Fukui, K., Maeda, K.: Face recognition using temporal image sequence. In: Proc. of AFGR, pp. 318–323 (1998)
10. Eyben, F., Wömer, M., Schuller, B.: Openear – introducing the Munich open-source emotion and affect recognition toolkit. In: Proc. 4th International HUMAINE Association Conference on Affective Computing and Intelligent Interaction 2009 (ACII 2009), Amsterdam, The Netherlands, vol. I, pp. 576–581. IEEE, Los Alamitos (2009)
11. Jain, A.K., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. Pattern Recognition 38, 2270–2285 (2005)

Listening-Test-Based Annotation of Communicative Functions for Expressive Speech Synthesis*

Martin Grůber and Jindřich Matoušek

Department of Cybernetics, Faculty of Applied Sciences,
University of West Bohemia, Pilsen, Czech Republic
{gruber,matousek}@kky.zcu.cz

Abstract. This paper is focused on the evaluation of listening test that was realized with a view to objectively annotate expressive speech recordings and further develop a limited domain expressive speech synthesis system. There are two main issues to face in this task. The first matter in issue to be taken into consideration is the fact that expressivity in speech has to be defined in some way. The second problem is that perception of expressive speech is a subjective question. However, for the purposes of expressive speech synthesis using unit selection algorithms, the expressive speech corpus has to be objectively and unambiguously annotated. At first, a classification of expressivity was determined making use of communicative functions. These are supposed to describe the type of expressivity and/or speaker's attitude. Further, to achieve objectivity at a significant level, a listening test with relatively high number of listeners was realized. The listeners were asked to mark sentences in the corpus using communicative functions. The aim of the test was to acquire a sufficient number of subjective annotations of the expressive recordings so that we would be able to create "objective" annotation. There are several methods to obtain objective evaluation from lots of subjective ones, two of them are presented.

Keywords: expressive speech synthesis, listening test, communicative functions, inter-rater agreement measure.

1 Introduction

Current speech synthesis techniques are surely able to produce high quality and intelligible speech. However, if we are talking about artificial speech that should not be recognized from human speech, some kind of speaker's attitude have to be considered and incorporated in the speech production process. It means that some expressivity or emotions in accordance with the content of speech will for sure improve the perception of the communicated information by listeners. Perhaps, this issue is not so hot in terms of some information systems or call centers which also use synthesized speech but in

* This work was partially funded by the Companions project IST-FP6-034434. This research was also supported by the Grant Agency of the Czech Republic, project No. GAČR 102/09/0989 and partly also by the University of West Bohemia, project No. SGC-2010-054. The access to the METACentrum computing facilities provided under the research intent MSM6383917201 is highly appreciated.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 283–290, 2010.

tasks dealing with personal dialogues between a computer and a human it should be taken into consideration.

The task of general expressive speech synthesis within unlimited domain is so extensive and complex that it is beyond present technical capabilities. Therefore we need to limit this task somehow. We are speaking on dialogues between a computer and a human but it is not restrictive enough. Our task was determined as a dialogue between a senior and a computer and theme for conversations was set to reminiscing about personal photographs. It will be shown that this way the domain is limited enough to improve our current speech synthesis and to create an expressive speech synthesizer.

Since our current TTS system ARTIC [1] is corpus oriented and based on unit selection algorithms [2], the improvement of speech synthesis consists in speech corpus enhancement. Thus an expressive speech corpus was recorded and annotated using various categories of expressivity by means of a listening test. Reliability of such annotation was proved using measures of inter-listeners agreement.

The paper is organized as follows: In Section 2 the expressive speech recording process is briefly described. The background, preparation works and settings of the performed listening test is shown in Section 3. In Section 4 we focused on the evaluation of the listening test with respect to credibility and reliability of the listeners. Finally, in Section 5, conclusions arising from the listening test results are drawn and future work is outlined.

2 On the Expressive Speech Corpus

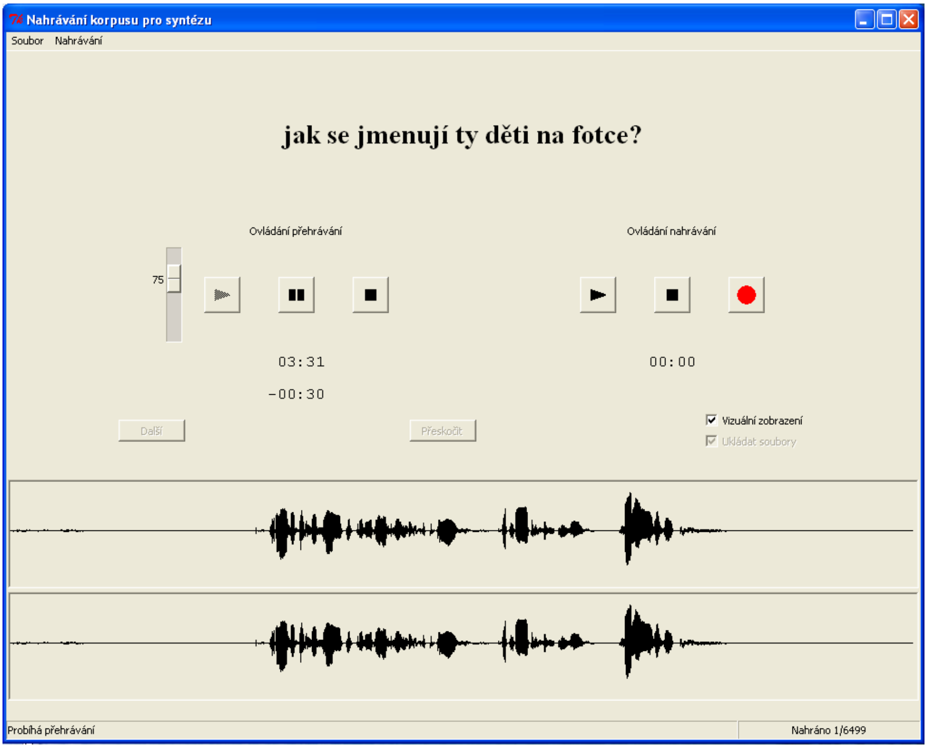
To incorporate expressivity into our current TTS system, an expressive speech corpus was recorded and merged together with existing neutral one. Issues in this task include but are not limited to corpus design, corpus recording, description of expressivity that is supposed to be contained in the corpus and annotation of the expressive recordings using the defined expressivity categories. These issues are more discussed in the following sections.

2.1 Corpus Design

Since we are dealing with limited domain expressive speech synthesis, definition of the domain is necessary. The domain in this task was restricted to dialogues between seniors and a computer. Theme for these conversations was set to reminiscing about seniors' photographs. This limitation is already sufficient enough. To become familiar with these conversations, an extensive audiovisual database containing natural dialogues between seniors and a computer (applying 3D avatar – "talking head" [3] with a neutral TTS system) was recorded using Wizard of Oz method and manually transcribed. Process of database recording is presented in [4]. On the basis of the recorded database we got knowledge about how the natural dialogues develop, what the seniors like to talk about and what kind of expressivity is expected to be conveyed within the synthesized speech.

2.2 Corpus Recording

We have decided to proceed with the expressive corpus creation as follows. First, we hired a professional female speaker (stage-player) and instructed her not to express a specific emotions but just to put herself in the place of a partner for seniors in a dialogue – pretend to be an avatar. In order to facilitate such an empathy, a special software application was developed (see Figure 1) – it played back the parts of the natural dialogues when the senior was speaking (to provide the speaker with the relevant context) and at the time when the avatar have originally spoken, the dialogue was paused and the speaker was prompted to record the avatar's sentence herself. The text of the actual sentence was displayed on the screen even when the real (context) dialogue was being played so that the speaker had enough time to get acquainted with it before the recording. Also time remaining to the recording of the next utterance was displayed. Controlling of the application was designed to be very easy for the speaker so that she could have been fully concentrated on the recording.



jak se jmenují ty děti na fotce?

03:31

-00:30

00:00

Probíhá přehrávání

Nahráno 1/6499

Fig. 1. Software interface for expressive corpus recording with the use of real dialogues

The recording equipment was carefully selected and set-up in order to ensure the highest possible technical quality of the corpus – the speaker was placed in the anechoic room and the recording was done using a professional mixing desk. The glottal signal was captured along with the speech. That way we have recorded more than 7,000 of (mostly short) sentences. Those were carefully transcribed.

2.3 Expressivity Description

The issue of expressivity description is very complex. In the past, several techniques were proposed and are divided into two main groups. One basic approach is to use continuous representation in two-dimensional space introduced in [5]; any kind of expressivity is referenced as a point with specific coordinates in that space. The other alternative is a categorical view; any kind of expressivity is classified into one (or more) of predefined classes.

For purposes of expressive speech synthesis and machine processing, the categorical classification of expressivity seems to be more suitable. Therefore we decided to utilize this approach. Moreover, since unit selection algorithms are applied, the expressivity class can be used as a feature for each particular unit which is stored in a unit inventory.

Specification of an appropriate set of classes for the limited domain defined above was based on dialogue acts proposed in [6]. This set was modified for our purposes and is shown in Table 1. Since each of the categories expresses function of a sentence in communication, it is called *communicative function*.

Table 1. Set of communicative functions

communication function	*symbol*	*example*
directive	DIRECTIVE	Tell me that. Talk.
request	REQUEST	Let's get back to that later.
wait	WAIT	Wait a minute. Just a moment.
apology	APOLOGY	I'm sorry. Excuse me.
greeting	GREETING	Hello. Good morning.
goodbye	GOODBYE	Goodbye. See you later.
thanks	THANKS	Thank you. Thanks.
surprise	SURPRISE	Do you really have 10 siblings?
sad empathy	SAD-EMPATHY	I'm sorry to hear that. It's really terrible.
happy empathy	HAPPY-EMPATHY	It's nice. Great. It had to be wonderful.
showing interest	SHOW-INTEREST	Can you tell me more about it?
confirmation	CONFIRM	Yes. Yeah. I see. Well. Hmm.
disconfirmation	DISCONFIRM	No. I don't understand.
encouragement	ENCOURAGE	Well. For example? And what about you?
not specified	NOT-SPECIFIED	Do you hear me well? My name is Paul.

2.4 Annotation

The expressive speech corpus was annotated using communicative functions by means of a listening test. The test was aimed to determine objective annotation on the basis of several subjective annotations as the perception of expressivity is always subjective and may vary depending on particular listener. Preparation works and listening test framework are described in the following section. Evaluation of listening test result and a measure of inter-rater agreement analysis is presented in Section 4.

3 Listening Test Background

The listening test was organized on the client-server basis using a specially developed web application. This way listeners were able to work on the test from their homes without any contact with the test organizers. The listeners were required to have only an internet connection, any browser installed on their computers and some device for audio playback. Various measures were undertaken to detect possible cheating, carelessness or misunderstandings.

Potential test participants were addressed mostly among university students from all faculties and the finished listening test was financially rewarded (to increase motivation for the listeners). The participants have been instructed to listen to the recordings very carefully and subsequently mark communicative functions that are expressed within the sentence. The number of possibly marked communicative functions for one utterance was just upon the listeners, they were not limited anyhow. Few sample sentences labelled with communicative functions were provided and available to the listeners on view at every turn. If any listener marked one utterance with more than one communicative function, he was also required to specify whether the functions occur in that sentence consecutively or concurrently. If the communicative functions are marked as consecutive in a particular utterance, this utterance is omitted from further research for the present. These sentences should be later manually reviewed and either divided into more shorter sentences or omitted completely.

Finally, 12 listeners have successfully finished the listening test. However, this way we obtained subjective annotations that vary across the listeners. To objectively annotate the expressive recordings, proper combination of the subjective annotations was needed. Therefore an evaluation of the listening test was made.

4 Listening Test Evaluation

4.1 Objective Annotation

We utilized two ways to deduce the objective annotation.

The first way is a simple majority method. Using this easy and intuitive approach, each sentence is assigned a communicative function, that was marked by the majority of the listeners. In case of less then 50% of all listeners marked such communicative function, the classification of this sentence is considered as untrustworthy.

The second approach is based on maximum likelihood method. Maximum likelihood estimation is a statistical method used for fitting a statistical model to data and providing estimates for the model's parameters. Under certain conditions, the maximum likelihood estimator is consistent. The consistency means that having a sufficiently large number of observations (annotations in our case), it is possible to find the value of statistical model parameters with arbitrary precision. The parameter calculation is implemented using the EM algorithm [7]. Knowing the model parameters we are able to deduce true observation which we call objective annotation. Precision of the estimate is one of the outputs of this model. Using the precision, any untrustworthy assignment of a sentence with a communicative function can be eliminated.

Comparing these two approaches, 35 out of 7287 classifications were marked as untrustworthy using maximum likelihood method and 571 using simple majority method. The average ratio of listeners who marked the same communicative function for particular sentence using simple majority approach was 81%, when untrustworthy classifications were excluded. Similar measure for maximum likelihood approach cannot be easily computed as the model parameters and the estimate precision depend on number of iteration in the EM algorithm.

We decided to use the objective annotation obtained by maximum likelihood method. It is an asymptotically consistent, asymptotically normal and asymptotically efficient estimate. We have also successfully used this approach in recent works regarding speech synthesis research, see [8].

Further, we need to confirm that the listeners marked the sentences with communicative functions consistently and achieved some measure of agreement. Otherwise the subjective annotations could be considered as accidental or the communicative functions inappropriately defined and thus the acquired objective annotation would be false. For this purpose, we make use of two statistical measures for assessing the reliability of agreement among listeners.

One of the measures used for such evaluation is Fleiss' kappa. It is a statistical measure for assessing the reliability of agreement between a fixed number of raters when assigning categorical ratings to a number of items or classifying items. We calculated this measure among all listeners separately for each communicative function. Computation of overall Fleiss' kappa is impossible because the listeners were allowed to mark more than one communicative function for each sentence. However, the overall value can be evaluated as the mean of Fleiss' kappas of all communicative functions.

Another measure used here is Cohen's kappa. It is a statistical measure of inter-rater agreement for categorical items and takes into account the agreement occurring by chance as well as Fleiss' kappa. However, Cohen's kappa measures the agreement only between two listeners. We decided to measure the agreement between each listener and the objective annotation obtained by maximum likelihood method. Again, calculation of Cohen's kappa was made for each communicative function separately. Thus we can find out whether particular listener was in agreement with the objective annotation for certain communicative function. Finally, the mean of Cohen's kappas of all communicative functions was calculated.

Results of agreement measures are presented in Table 2. Value of Fleiss' and Cohen's kappa vary between 0 and 1, the higher value the better agreement. More detailed interpretation of measure of agreement is in [9].

The Fleiss' kappa mean value of 0.5434 means that the measure of inter-listeners agreement is moderate. As it is obvious from Table 2, communicative functions *OTHER* and *NOT-SPECIFIED* should be considered as poorly recognizable. It is understandable when taking into consideration their definitions. After eliminating values of these communicative functions the mean value of 0.6191 is achieved, which means substantial agreement among the listeners.

The Cohen's kappa mean value of 0.6632 means that the measure of agreement between listeners and objective annotation is substantial. Moreover, we can again eliminate communicative functions *OTHER* and *NOT-SPECIFIED* as they were poorly

Table 2. Fleiss' and Cohen's kappa and occurrence probability for various communicative functions and for the "consecutive CFs" label. For Cohen's kappa, mean value and standard deviation is presented, since Cohen kappa is measured between annotation of each listener and the reference annotation.

communication function	Fleiss's kappa	Measure of agreement	Cohen's kappa	Cohen's kappa SD	Measure of agreement	Occurr. probab.
DIRECTIVE	0.7282	Substantial	0.8457	0.1308	Almost perfect	0.0236
REQUEST	0.5719	Moderate	0.7280	0.1638	Substantial	0.0436
WAIT	0.5304	Moderate	0.7015	0.4190	Substantial	0.0073
APOLOGY	0.6047	Substantial	0.7128	0.2321	Substantial	0.0059
GREETING	0.7835	Substantial	0.8675	0.1287	Almost perfect	0.0137
GOODBYE	0.7408	Substantial	0.7254	0.1365	Substantial	0.0164
THANKS	0.8285	Almost perfect	0.8941	0.1352	Almost perfect	0.0073
SURPRISE	0.2477	Fair	0.4064	0.1518	Moderate	0.0419
SAD-EMPATHY	0.6746	Substantial	0.7663	0.0590	Substantial	0.0344
HAPPY-EMPATHY	0.6525	Substantial	0.7416	0.1637	Substantial	0.0862
SHOW-INTEREST	0.4485	Moderate	0.6315	0.3656	Substantial	0.3488
CONFIRM	0.8444	Almost perfect	0.9148	0.0969	Almost perfect	0.1319
DISCONFIRM	0.4928	Moderate	0.7153	0.1660	Substantial	0.0023
ENCOURAGE	0.3739	Fair	0.5914	0.3670	Moderate	0.2936
NOT-SPECIFIED	0.1495	Slight	0.3295	0.2292	Fair	0.0736
OTHER	0.0220	Slight	0.0391	0.0595	Slight	0.0001
mean	*0.5434*	*Moderate*	*0.6632*		*Substantial*	
consecutive CF	0.5138	Moderate	0.6570	0.2443	Substantial	0.0374

recognizable also according to Cohen's kappa. Thus, mean value of 0.7316 is achieved. However, it is still classified as substantial agreement.

As it is shown in Table 2, agreement among listeners regarding classification of consecutive communicative function was measured too. The listeners agreed on this label moderately among each other and substantially with the objective annotation. There are also shown probabilities of the particular communicative functions occurrence when maximum likelihood method was used for the objective annotation obtaining. It is obvious that communicative functions *SHOW-INTEREST* and *ENCOURAGE* are the most frequent.

5 Conclusion and Future Work

In this work we have created an objectively annotated expressive speech corpus. The subjective annotations of expressivity was made by means of listening test, where listeners marked each sentence from the corpus with communicative functions. The objective annotation was deduced from the subjective ones using maximum likelihood method. The inter-listeners measures of agreement confirmed that the objective annotation is trustworthy.

Appropriate combination of the expressive speech corpus and current neutral corpus will allow us to create an expressive speech synthesizer. Its development is our objective for future work. The synthesizer is planned to be used in a limited domain dialogue system, which is going to serve elderly people to discuss their personal photographs with computer. We should also deal with social issues regarding such a human-computer interaction.

References

1. Matoušek, J., Tihelka, D., Romportl, J.: Current State of Czech Text-to-Speech System ARTIC. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 439–446. Springer, Heidelberg (2006)
2. Tihelka, D., Romportl, J.: Exploring Automatic Similarity Measures for Unit Selection Tuning. In: Proceedings of Interspeech, pp. 736–739. ISCA, Brighton (2009)
3. Železný, M., Krňoul, Z., Císař, P., Matoušek, J.: Design, Implementation and Evaluation of the Czech Realistic Audio-visual Speech Synthesis. Signal Processing 12, 3657–3673 (2006)
4. Grůber, M., Legát, M., Ircing, P., Romportl, J., Psutka, J.: Czech Senior COMPANION: Wizard of Oz Data Collection and Expressive Speech Corpus Recording. In: Human Language Technologies as a Challenge for Computer Science and Linguistics, pp. 266–269. Wydawnictvo Poznanskie, Poznan (2009)
5. Russel, J.A.: A Circumplex Model of Affect. Journal of Personality and Social Psychology 39, 1161–1178 (1980)
6. Syrdal, A.K., Kim, Y.-J.: Dialog Speech Acts and Prosody: Considerations for TTS. In: Proceedings of Speech Prosody, pp. 661–665. Campinas, Brazil (2008)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B 39(1), 1–38 (1977)
8. Romportl, J.: Prosodic Phrases and Semantic Accents in Speech Corpus for Czech TTS Synthesis. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 493–500. Springer, Heidelberg (2008)
9. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics 33(1), 159–174 (1977)

Czech HMM-Based Speech Synthesis*

Zdeněk Hanzlíček

Department of Cybernetics, University of West Bohemia,
Univerzitní 8, 306 14, Pilsen, Czech Republic
zhanzlic@kky.zcu.cz
http://www.kky.zcu.cz/en

Abstract. In this paper, first experiments on statistical parametric HMM-based speech synthesis for the Czech language are described. In this synthesis method, trajectories of speech parameters are generated from the trained hidden Markov models. A final speech waveform is synthesized from those speech parameters. In our experiments, spectral properties were represented by mel cepstrum coefficients. For the waveform synthesis, the corresponding MLSA filter excited by pulses or noise was utilized. Beside that basic setup, a high-quality analysis/synthesis system STRAIGHT was employed for more sophisticated speech representation. For a more robust model parameter estimation, HMMs are clustered by using decision tree-based context clustering algorithm. For this purpose, phonetic and prosodic contextual factors proposed for the Czech language are taken into account. The created clustering trees are also employed for synthesis of speech units unseen within the training stage. The evaluation by subjective listening tests showed that speech produced by the combination of HMM-based TTS system and STRAIGHT is of comparable quality as speech synthesised by the unit selection TTS system trained from the same speech data.

Keywords: HMM-based speech synthesis, TTS, Czech language.

1 Introduction

Nowadays beside concatenative unit selection method, HMM-based speech synthesis [1] is one of the most researched synthesis methods. In this synthesis method, hidden Markov models (or possibly other parametric models) are trained from natural speech database. Spectral parameters, fundamental frequency, duration and eventually some excitation parameters are modelled simultaneously by the corresponding HMMs. For a more robust model parameter estimation, HMMs are clustered by using decision tree-based context clustering algorithm. For this purpose, phonetic and prosodic contextual factors are taken into account. They respect phonetic, prosodic and linguistics characteristics of given language. The created clustering trees are also employed for synthesis

* This work was supported by the Grant Agency of the Czech Republic, project No. GAČR 102/09/0989 and by the Ministry of Education of the Czech Republic, project No. 2C06020. Author would also like to thank Prof. Hideki Kawahara from Wakayama University for his permission to use the STRAIGHT analysis/synthesis method [2]. The access to the MetaCentrum supercomputing facilities provided under the research intent MSM6383917201 is also highly appreciated.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 291–298, 2010.

of speech units unseen within the training stage. During synthesis, trajectories of all speech parameters are generated from these trained models in the maximum likelihood sense. A final speech waveform is reconstructed from those speech parameters.

Originally, this method was developed for Japanese. However, as well as other synthesis methods, it is mainly language independent. Thus, TTS systems for many other languages (e.g. English [3]) have been successfully implemented. For a more detailed language listing see e.g. [1]. The expansion of that synthesis method was possible thanks to the HTS toolkit [4] that provides statistical methods for HMM manipulation (including training, context clustering, parameter trajectory generation etc.).

This paper describes first experiments on statistical parametric HMM-based speech synthesis for the Czech language. For building of our experimental TTS system, we also employed the HTS toolkit. Within our experiments, two different speech analysis/synthesis methods and speech representations were compared:

1. A simple representation by Mel cepstrum coefficients. Methods for parameter extraction and speech synthesis by using MLSA filter were provided by SPTK toolkit [5].
2. More sophisticated speech representation by the high-quality analysis/synthesis method STRAIGHT [2]. It has been already used in HMM-based speech synthesis framework, e.g. by Zen et al. [6].

A large listening test was organized for the evaluation and comparison of various settings of our experimental TTS system. Results showed that speech produced by the combination of HMM-based TTS system and STRAIGHT is of comparable quality as speech synthesised by the unit selection TTS system trained from the same speech data.

The paper is organized as follows. Section 2 deals with the phonetic and prosodic characteristics of the Czech language. According to them, a suitable set of contextual factors is proposed for the purposes of Czech HMM-based speech synthesis. In Section 3 a brief description of HMM-based speech synthesis system and its settings for our experiments are presented. Results of performed listening tests are shown in Section 4. Finally, Section 5 summarizes the paper and outlines our future work.

2 Czech Language Characteristics

For naturally sounding synthesized speech, phonetic and prosodic characteristics of a particular language should be taken into account. In HMM-based speech synthesis method, these language characteristics are respected by definition of so called contextual factors. Then, a speech unit is given as a phoneme with its phonetic and prosodic context information. In this manner, the language prosody is implicitly modelled, because for various contexts different units/models can be used. Contextual factors for the Czech language are summarized in Section 2.3.

2.1 Phonetic Characteristics

The set of Czech phones is defined in Table 1. In our experiments, phonemes from the basic set were used. In addition we also employed glottal stop `[?]`, inter-word pause

[#] and long silence [$]. Other allophones listed in Table 1 could also be utilised. However they usually correspond to basic phonemes in a special phonetic context, thus in a system with context depended units, allophones and basic phonemes are implicitly distinguished.

Table 1. Czech phonetic inventory used in our TTS system (in SAMPA [7] notation)

Basic Set	Vowels	[a], [a:], [e], [e:], [i], [i:], [o], [o:], [u], [u:]
	Diphthongs	[o_u], [a_u], [e_u]
	Plosives	[p], [b], [t], [d], [c], [J\], [k], [g]
	Nasals	[m], [n], [J]
	Fricatives	[f], [v], [s], [z], [Q\], [P\], [S], [Z], [x], [h], [j]
	Liquids	[r], [l]
	Affricates	[t_s], [d_z], [t_S], [d_Z]
Allophones		[F], [N], [?], [G], [r=], [l=], [m=], [@]

Sometimes, the syllable is considered to be an alternative phonetic unit to the phone in the Czech language. Syllabification and syllable utilisation in context of concatenative speech synthesis were researched e.g. in [8]. Though, they seem not to be suitable basic units for speech synthesis, the information on syllable boundaries in text should be taken into account, because it has obviously some influence on (human) speech production.

2.2 Prosodic Characteristics

The prosodic structure of the Czech language can be described with the prosodic phrase grammar [9,10] which defines a hierarchical tree structure above a synthesised utterance. The following functionally relevant structures are distinguished:

- *Prosodic sentence* – syntactically consistent unit that usually corresponds to the whole utterance.
- *Prosodic clause* – linear unit in speech delimited by pauses.
- *Prosodic phrase* – segment of speech containing a certain continuous intonation scheme.
- *Prosodeme* – a rather abstract unit describing communication function. In the Czech language, this function is usually connected with the last prosodic word in the phrase. For other prosodic words, a formal null prosodeme is defined. For the complete prosodeme description see [9].
- *Prosodic word* – group of words belonging to one stress, often considered as a basic rhythmic unit.
- *Semantic accent* – emphasis of a prosodic word.

For the present, the prosodic phrases and semantic accents were not employed in our experimental setup. Their detection and modelling is not an easy task. However, a statistical method for their assignment in speech data has been already developed and described in [11]. Application of prosodic phrases and semantic accents is planned in future experiments.

2.3 Contextual Factors

Regarding the characteristics of the Czech language, a set of suitable contextual factors was defined. It is presented in Table 2. In comparison with other languages, e.g. English [3], this set is very reduced. However, greater amount of contextual factors or also greater amount of their possible values result in higher computational requirements. Thus within our primary experiments, we decided for a reduced set of factors. Significance of particular contextual factors is planned to be researched in the future.

Table 2. Contextual factors

Factor	Possible values
Previous and next phoneme	see Table 1
Phone location in syllable	first, inner, last, single
Syllable location in prosodic word	
Prosodic word location in clause	
Clause location in sentence	
Prosodeme type	terminating satisfactorily, terminating unsatisfactorily, nonterminating, null

3 HMM-Based TTS System

A thorough description of an HMM-based TTS system, including utilised statistical methods, appeared in many publications, e.g. [1]. This section gives only a brief overview, because these methods are not the object of our contribution. For building of our experimental HMM-based TTS system, the following tools were utilised

- tools for speech analysis and reconstruction
  - SPTK - Speech Signal Processing Toolkit [5]
  - STRAIGHT - Speech Transformation and Representation based on Adaptive Interpolation of weiGHTed spectrogram [12]
- tools for HMM manipulation (training, parameter generation etc.)
  - HTS - HMM-based Speech Synthesis System [4]

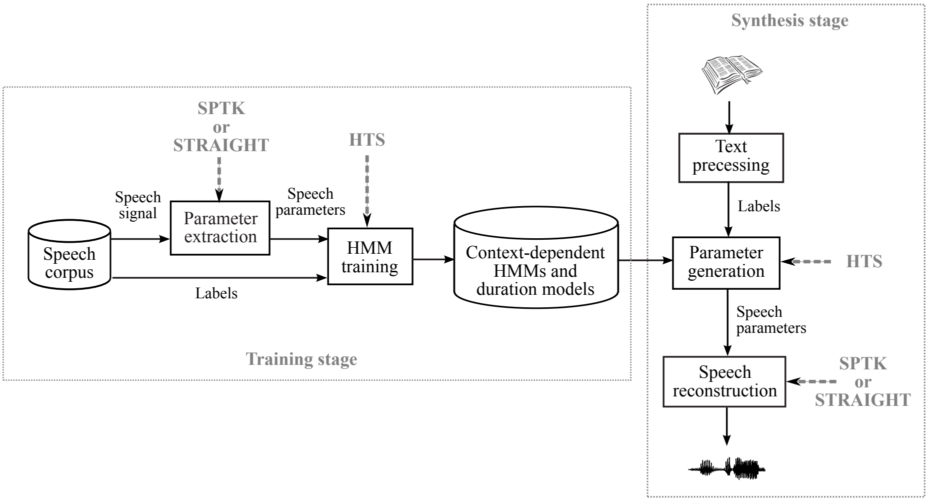


Fig. 1. A simplified scheme of our HMM-based TTS system

Training stage can be roughly divided into 3 main parts:

1. **Parameter extraction** – speech signal was sampled at 16 kHz.
   - For the mel cepstral analysis a 25 ms Blackman window with 5 ms shift was employed. Each speech frame was represented by a composed parameter vector containing 25 mel cepstral coefficients and F_0 value with their delta and delta-delta.
   - The STRAIGHT analysis method used Gaussian F_0 adaptive window with 5 ms shift. Composed parameter vector contained 40 mel cepstral coefficients, F_0 value and 5 aperiodicity coefficients, again with their delta and delta-delta.
2. **Model training** – model parameters are estimated from speech data by using maximum likelihood criterion. First, robust models for particular single phoneme are trained. Then, models for all particular combinations of contextual factors within training data are estimated.
3. **Context clustering** – for a more robust model parameter estimation, clustering of contextual factors is performed.

In the synthesis stage, trajectories of speech parameters are generated directly from the trained HMMs. Clustering trees from the training stage are utilised to find a suitable substitute for units unseen in training data. The final speech waveform is reconstructed from the generated parameters by using appropriate synthesis methods, i.e. MLSA filter excited by pulses/noise or STRAIGHT-based vocoding.

4 Experiments and Results

For our experiments, two different voices (male and female) were employed. Both were professional speakers with broadcast experiences. Speech data were originally recorded for purposes of a concatenative TTS system utilizing unit selection method [13].

One large MOS (mean opinion score) listening test was conducted to evaluate the quality of speech produced by our experimental TTS system. In this test, all combinations of speaker (male or female), speech representation (by using SPTK or STRAIGHT) and amount of training data (10 minutes, 1 hour and 5 hours without pauses) appeared. In addition, natural utterances from source speech corpus and utterances produced by a concatenative TTS system were also mixed in this test. The concatenative TTS system [10], that employed unit selection synthesis method, was trained from the same speech data.

18 listeners took part in this test, they listened to single utterances (96 sentences in sum) and evaluated them according to the overall quality (acceptance) by using the standard MOS scale:

1. bad quality
2. poor quality
3. fair quality
4. good quality
5. excellent quality

Table 3. MOS test results (mean score ± standard deviation)

Speech generation method	Training data amount	Notation	Score	
			Male	Female
Natural speech		NAT	4.76 ± 0.06	4.04 ± 0.19
MLSA + pulses/noise	10 minutes	M–10m	2.20 ± 0.24	1.70 ± 0.29
	1 hour	M–1h	2.62 ± 0.34	2.18 ± 0.31
	5 hours	M–5h	2.73 ± 0.39	2.29 ± 0.48
STRAIGHT	10 minutes	S–10m	2.75 ± 0.21	1.68 ± 0.34
	1 hour	S–1h	3.56 ± 0.24	2.83 ± 0.15
	5 hours	S–5h	3.83 ± 0.28	3.38 ± 0.31
Unit selection		US	3.57 ± 0.62	3.14 ± 0.66

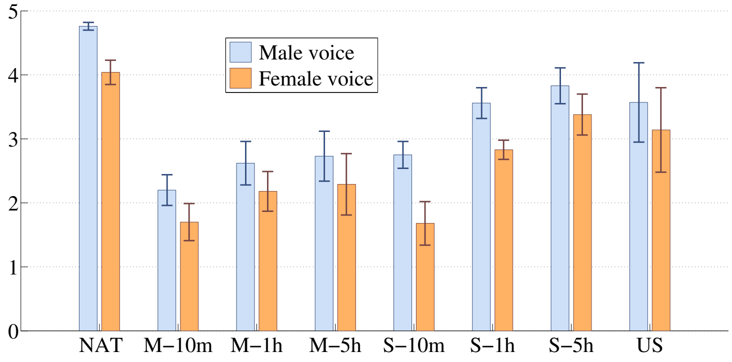


Fig. 2. MOS test results

The results of that test are presented in Table 3 and Figure 2. Expectably, speech quality increased with the amount of training data. A complex excitation modelling by STRAIGHT proved also very significant for quality perception. These findings are in accordance with other research works, e.g. [6]. For a comparable amount of training data HMM-based synthesis system with STRAIGHT produces speech of similar quality as system with unit selection method.

5 Conclusion

In this paper, first experiments on statistical parametric HMM-based speech synthesis for the Czech language were presented. For building an experimental TTS system, HTS toolkit was utilised. For speech representation, two different speech analysis/synthesis methods were used: Mel cepstral analysis + synthesis by using MLSA filter and STRAIGHT. The evaluation by subjective listening tests showed that speech produced by the HMM-based TTS system with STRAIGHT is of comparable quality as speech synthesised by the unit selection TTS system trained from the same speech data.

5.1 Future Work

In our future experiments, we will mainly focus on two important task:

- *Contextual factors* - in our experiments only a simple set of contextual factors was employed. The influence of other prosodic and linguistics characteristics of the Czech language (e.g. prosodic phrase and semantic accent) should be also analysed.
- *Excitation representation* - the comparison of MLSA filter excited by pulses or noise and STRAIGHT synthesis method confirmed that the proper excitation modelling has a significant influence on resulting speech quality. Thus we plan to research more methods and models for excitation representation (e.g. ML excitation method [14]).

References

1. Zen, H., Tokuda, K., Black, A.W.: Statistical Parametric Speech Synthesis. Speech Communication 51, 1039–1064 (2009)
2. Kawahara, H., Masuda-Katsuse, I., de Cheveigne, A.: Restructuring Speech Representations using a Pitch-Adaptive Time-Frequency Smoothing and an Instantaneous-Frequency-based F0 Extraction: Possible Role of a Repetitive Structure in Sounds. Speech Communication 27, 187–207 (1999)
3. Tokuda, K., Zen, H., Black, A.W.: An HMM-based Speech Synthesis System Applied to English. In: Proc. of IEEE Workshop on Speech Synthesis, pp. 227–230 (2002)
4. HMM-based Speech Synthesis System (HTS), `http://hts.sp.nitech.ac.jp`
5. Speech Signal Processing Toolkit (SPTK), `http://sp-tk.sourceforge.net`
6. Zen, H., Toda, T., Nakamura, M., Tokuda, K.: Details of the Nitech HMM-Based Speech Synthesis System for the Blizzard Challenge 2005. IEICE Transactions on Information and Systems E90-D, 325–333 (2007)
7. Czech SAMPA, `http://www.phon.ucl.ac.uk/home/sampa/czech-uni.htm`

8. Matoušek, J., Hanzlíček, Z., Tihelka, D.: Hybrid Syllable/Triphone Speech Synthesis. In: Proc. of Interspeech 2005, Lisbon, Portugal, pp. 2529–2532 (2005)
9. Romportl, J., Matoušek, J., Tihelka, D.: Advanced Prosody Modelling. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 441–447. Springer, Heidelberg (2004)
10. Tihelka, D., Matoušek, J.: Unit Selection and its Relation to Symbolic Prosody: A New Approach. In: Proc. of Interspeech 2006 – ICSLP, Pittsburgh, Pennsylvania, vol. 1, pp. 2042–2045 (2006)
11. Romportl, J.: Prosodic Phrases and Semantic Accents in Speech Corpus for Czech TTS Synthesis. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 493–500. Springer, Heidelberg (2008)
12. STRAIGHT, a speech analysis, modification and synthesis system, http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTadv/index_e.html
13. Matoušek, J., Romportl, J.: Recording and Annotation of Speech Corpus for Czech Unit Selection Speech Synthesis. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 326–333. Springer, Heidelberg (2007)
14. Maia, R., Toda, T., Zen, H., Nankaku, Y., Tokuda, K.: An Excitation Model for HMM-Based Speech Synthesis Based on Residual Modelling. In: Proc. of 6th ISCA Workshop on Speech Synthesis, pp. 131–136 (2007)

Using Syllables as Acoustic Units for Spontaneous Speech Recognition

Jan Hejtmánek

Laboratory of Intelligent Communication Systems,
Dept. of Computer Science and Engineering,
University of West Bohemia in Pilsen, Czech Republic
hejtman2@kiv.zcu.cz

Abstract. In this work, we deal with advanced context-dependent automatic speech recognition (ASR) of Czech spontaneous talk using hidden Markov models (HMM). Context-dependent units (e.g. triphones, diphones) in ASR systems provide significant improvement against simple non-context-dependent units. However, for spontaneous speech recognition we had to overcome some very challenging tasks. For one, the number of syllables compared to the size of spontaneous speech corpus makes the usage of context-dependent units very difficult. The main part of this article shows problems and procedures to effectively build and use a syllable-based ASR with the LASER (ASR system developed at Department of Computer Science and Engineering, Faculty of Applied Sciences). The procedures are usable with virtual any modern ASR.

1 Introduction

This document describes the way how to effectively use syllables as a context-dependent phonetic unit in automatic speech recognition of spontaneous speech. As we have shown in previous works [2,3] using context-dependent units (e.g. triphones or syllables) can lead to improvement in over-all recognition accuracy.

For this work we concentrated on spontaneous speech recognition using syllables as this aproach gets more and more attention ([5,7,8]). The speaker inserts many faults into the spontaneous speech making it more difficult to automatically recognize. However, these faults and errors are very simmilar syllables. Then, according to our hypothesis, the syllables-based acoustic units should ensure better performance than triphones.

2 Syllables

From phonology definition, the syllable is a basic unit of pronunciation that consists of a central syllabic element (usually a vowel), and can be preceded and/or followed by none or more consonants. The central syllabic element is called nucleus, consonants that precede nucleus are onset, and consonants that follow nucleus are coda.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 299–305, 2010.

2.1 Distiguishing Syllables

The structure of syllables is a combination of allowable segments and typical sound sequences (which are language specific). These segments are shown in Figure 1 with the example of English word limit. The segments are made from consonants (C) and vowels (V). We distinguish four basic types of syllables.

- Heavy syllables Has a branching rhyme. All syllables with a branching nucleus (long vowels) are considered heavy. Some languages treat syllables with a short vowel (nucleus followed by a consonant (coda) as heavy.
- Light syllables Has a non-branching rhyme (short vowel). Some languages treat syllables with a short vowel (nucleus) followed by a consonant (coda) as light.
- Closed syllables Syllables end with a consonant coda.
- Open syllables Has no final consonant.

Using the knowledge of how the syllables are created and used and a suf- ficiently large dictionary it is possible to gather nearly all used syllables in a language. Moreover, this statistics (made during the gathering of the dictionary) can be used to create a statistically built decision-tree.

2.2 Syllabification

Syllabification is the separation of a word (either spoken or written) into syllables. It has very strict rules with many exceptions. The process can be rule-based, pattern-based or statistic.

In previous works we examined several algorithms for syllabification of written language. Because the LASER is ASR for Czech spoken language, we further worked only on the Czech syllabification process. However, during the development and creation of Czech spontaneous ASR we confirmed that using only the spontaneous speech corpora is not enough to create a successful ASR (even for monophones). For this we adapted our syllabification algorithm to English with very good results.

Improved Lansky algorithm. Thanks to [1] we managed to obtain working basis for English and Czech syllabification. It is based on separating the vowels and creating the syllables by adding consonants to these vowels (vowels group). The process is very similar to a naive algorithm but it differs in the separation of consonants to the vowel groups:

1. Everything after the last vowel (vowel group) belongs to the last syllable
2. Everything before the first vowel (vowel group) belongs to the first syllable
3. If the number of consonants between vowels is even ($2n$), they are divided into the halves first half belongs to the left vowel(s) and second to the right vowel(s) (n/n).
4. If the number of consonants between vowel(s) is odd ($2n+1$), we divide them into $n/n+1$ parts.
5. If there is only one consonant between vowels, it belongs to the left vowel(s).

This method has several benefits. It is very fast using a regular expressions we can syllabificate in a fractions of real-time. Seccond, we get over 93% accuracy. When not accurate, the pseudo-syllable nevertheless complies with the syllable rules and the errors are made in parts of words where the human speaker makes them.

3 Speech Corpora

For our tests, two corpora were used. First, the train corpus, the set of read train station sayings. Seccond, our new spontaneous speech LACS corpus. This corpus is a set of spontaneous discourses taken during classes. These two corpora are compared in the Table 1.

Table 1. Corpora coparison

	Trains	LACS
All words	154,043	145,246
All distinct words	1,461	3,980
Number of speakers	30F/32M	14M
Total text size	2MB	1MB
Distinct syllables	829	2,504
Distinct triphones	2,268	5,906

The comparison shows that the LACS corpus is by far more complex than the other one.

4 Using Syllables in the ASR

The LASER uses internal configuration file structure compatible to the one HTK[1] uses (see [4]). Neither HTK nor LASER had the direct support for working with syllables and we had to implement a transformation algorithm. This only transforms configuration files for monophone ASR into the form for syllable ASR.

The biggest problem of the triphones is the number of the acoustic units (and therefore the number of the parameters of HMM]. To train such a huge number of units the training corpus has to be very large. The number of syllables (whe comparing to triphones) is much lower, hower the total number is still too high to be used with our training data.

To lower the number of parameters for training, the standard clustering technique are used. For our tests, we used data-driven and decision-tree clustering. Questions for the decision-tree are built using expert knowledge on creation of syllables.

5 Models and Clustering

To use the syllables in the HTK (LASER) recognizer it was necessary to adapt the models. First, the new models were built by concatenating monophone models to syllables. Thus models with variable number of states were created. These models will be referred to as “Syllables var”. For illustration see Figure 1.

[1] Hidden Markov Model Toolkit.

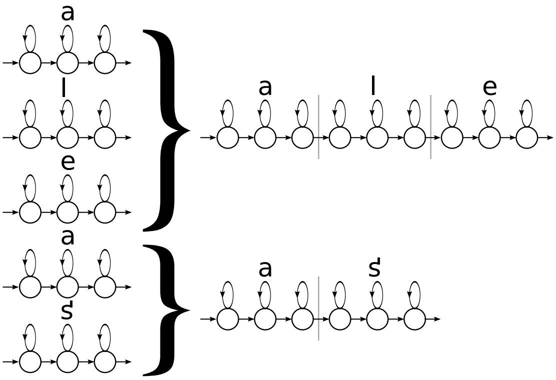


Fig. 1. Variable length of HMM

The monophone model is based on five-state HMM from which three states are emiting. Since the most common syllables in the train corpus are the twocharacter syllables, we build up the second testing model based on seven state HMM (with five emiting states). These models will be refered to as "Syllables 5". Results of ASR using various HMM is shown in the Figure 2.

Unlike other works [9,10,12,12], we use data-driven and decision-tree clustering. The data-driven clustering of syllables is straightforward. Clustering using decision-tree

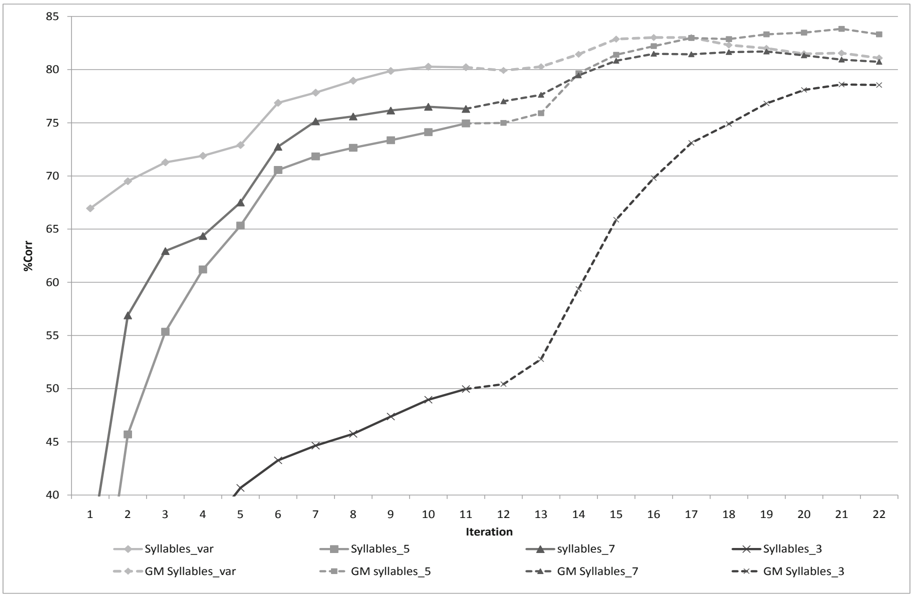


Fig. 2. ASR results with different HMM no. states

is more problematic. Thanks to the maping from monophones (Figure 1) we can apply very simmilar clustering method to the decision-tree based clustering for triphones.

Using statistics we group the syllables by their occurences in the syllable dictionary and their simillarity. The leaves are built to have minimum of 100 occurences in the training data. When taken in the proces of building of the decisiontree questions the syllable is grouped using two basic rules.

1. If the number of ocurences is lower than 50, the syllable cannot have separate leave.
2. If the number of occurrences is hihger than 100, the syllable cannot be grouped with any other like this.

The constants were taken from the statistics of syllables. 50 occurences is the threshold of 65% of units. 100 occurences corresponds to the top 15% of syllables.

Finding the best thresholds and its automation is the next step in our research.

Decision-tree clustering depends on the underlying model. The basic problem with static number of states is that we cannot know which states belongs to which part of syllable. variable number of states solves this.

6 Tests

6.1 Test Setup

The baseline test for both models is twelve iterations of training and testing using monophones. After this test we add Gaussian mixtures. Two mixtures are added in every iteration up to 32. The “Syllables var” models were then tested with data-driven clustering in HTK with thresholds 50, 100. The same thresholds were used for decision-tree clustering.

As we dont have the language model for Czech spontaneous speech yet, the language model was not used in any of the tests.

All the tests were made on Intel C2D 6700 CPU, 4 GB RAM, Windows XP Professional, HTK 3.3.

6.2 Comparison and Measurements

We use three basic measures to compare results Corr (Correct hits, in per- cents), Acc (Accuracy, in percents) and time (training and testing parts of every iteration, in percents).

6.3 Test Results

First, we tested different types of syllable-based acoustic models (2) on the Train corpus. The best performed Syllables 5. However to use the decision-tree clustering we used the Syllables var for the rest of the test. The performance in the first test is only slightly worse then the best Syllables 5.

From the previous work we knew that the syllable-based ASR with datadriven clustering is only slightly worse than the best triphone-based ASR. When we used the new acoustic models and clustering techniques with the read speech the decision-tree based ASR wins only by a small margin that is not statistically significant. Based on the references we stepped to the next tests the spontaneous speech.

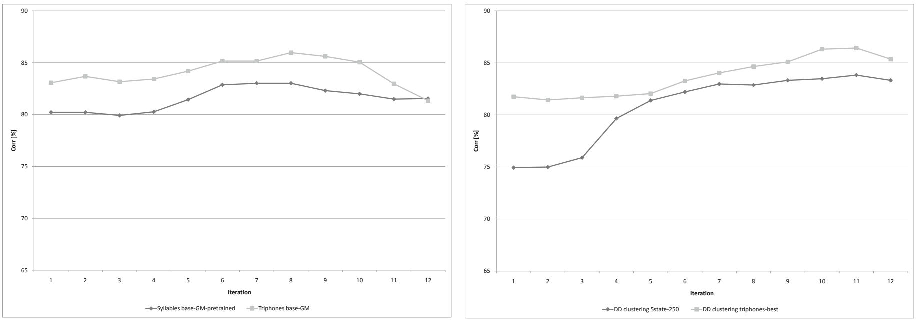


Fig. 3. Spontaneous speech (LACS) test results comparison

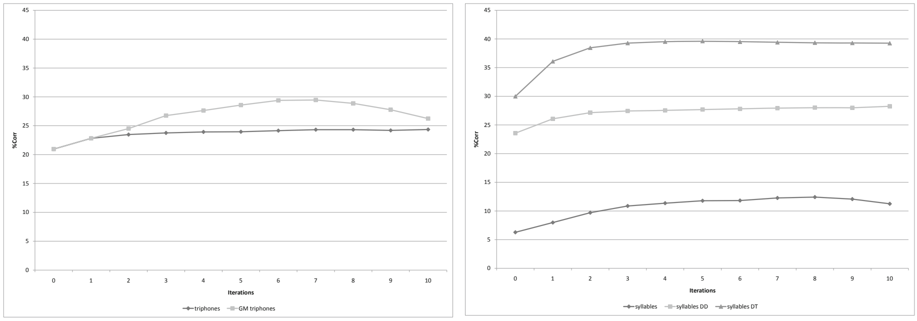


Fig. 4. Spontaneous speech (LACS) test results comparison

6.4 LACS Corpus Tests

We compared the results of test on the Train corpus (Figure 3). The cross-validation proved that the syllable-based and triphone-based ASR performance is within statistical error and we cannot declare that any of the acoustic model is better.

Results in the Figure 4 show the performance of triphone-based and syllable-based ASR on spontaneous speech corpus with no language model. The syllabel-based ASR (with decision-tree clustering) clearly outperforms the triphone-based by 10%. Moreover, the convergence of ASR during training proces is much faster.

7 Conclusion

We have successfully built and tested several syllable-based ASR systems. Thanks to contex-dependency we can model the language on the very low level of the ASR system. The results are very promising; however we need to verify them on a bigger corpus. As the algorithms are truly language-independent (with the syllable-based languages) we are preparing to test the algorithms on the VoxForge2 open source corpus. This corpus however doesnt include the spontaneous speech parts. For this we need to obtain a corpus for testing.

Acknowledgments. This work was supported by grant no. 2C06009 Cot-Sewing.

References

1. Lánský, J., Žemlička, M.: Text Compression: Syllables. In: Proceedings of the Dateso 2005 Annual International Workshop on Databases, Texts, Specifications and Objects. CEUR-WS, vol. 129, pp. 32–45 (2005)
2. Hejtmánek, J.: Use of Context-Dependent Units in Speech Recognition. Master thesis, University of West Bohemia in Pilsen, Faculty of Applied Sciences (2007)
3. Hejtmánek, J., Pavelka, T.: Use of Context-Dependent Units in Czech Speech. In: Proc. of Ph.D. Workshop 2007, Balatonfred, Hungary (2007)
4. Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.3), Cambridge University Engineering Department (2005)
5. Yu, K., Mason, J., Oglesby, J.: Speaker Recognition Models. In: Proceedings of Eurospeech 1995, pp. 629–632 (1995)
6. Laurinčukaité, S., Lipeika, A.: Syllable-Phoneme Based Continuous Speech Recognition. Institute of Mathematics and Informatics, Vilnius (2006)
7. Chang, S.: A Syllable, Articulatory-Feature and Stress-Accent Model of Speech Recognition. Berkeley. Ph.D. thesis. International Computer Science Institute (2002)
8. Ananthakrishnan, S., Narayanan, S.: Improved Speech Recognition Using Acoustic and Lexical Correlates of Pitch Accent in a N-best Rescoring Framework. Speech Analysis and Interpretation Laboratory Department of Electrical Engineerig Viterbi School of Engineering University of Southern California, Los Angeles (2007)
9. Chen, K., Hasegawa-Johnson, M., Cohen, A.: An automatic Prosody Labeling System Using ANN-based Syntactic-Prosodic Model and GMM-Based Acoustic-Prosodic Model. In: International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 509–512 (2004)
10. Han, Y., Boves, L.: EM Algorithm with Split and Merge in Trajectory Clustering for Automatic Speech Recognition. Department of Language and Speech, Radboud University Nijmegen (2006)
11. Shafran, I., Ostendorf, M.: Acoustic Model Clustering Based on Syllable Structure. Washington, Department of Electrical Engineering (2002)
12. SIL International, Glosary of linguistic Terms (2008), http://www.sil.org

Embedded Speech Recognition in UPnP (DLNA) Environment

Jozef Ivanecký and Radek Hampl

European Media Laboratory,
Schloss-Wolfsbrunnenweg 33, 69118 Heidelberg, Germany
{jozef.ivanecky,radek.hampl}@eml.org

Abstract. In the past decade great technological advances have been made in internet services, personal computers, telecommunications, media and entertainment. Many of these advances have benefited from sharing technologies across those industries. This influences how Digital Home Entertainment products are designed to follow the overall "Media Convergence" trend. Existing Universal Plug and Play (UPnP) or DLNA specifications are often used for these purposes. These specifications permit electronic devices to be simply plugged into home and local networks for access and exchange of shared data like music, video or photos. The number of media items in a user library can then easily exceed 10,000 elements. In addition, these specifications are used by manufacturers of consumer electronics to ensure interoperability of different consumer electronic devices.

In this paper, we describe our efforts towards introducing speech recognition to control electronic devices in UPnP (DLNA) environments. We give an overview of the content structure and media information available in the UPnP (DLNA) network. We also analyze the use of available information for speech recognition. The main focus will be on the possibility of designing and implementing a voice-enabled UPnP (DLNA) Control Point, and the introduction of one particular solution.

1 Introduction

In home entertainment today terms like MP3, AVI, MPEG, etc., appear very often. They indicate not only a new way of storing music, movies or pictures, but also gaining instant access to the entire music, video or picture selection no matter where they are stored. Distributed media content, as well as demand for interoperability among devices from different manufacturers, led to the Universal Plug And Play (UPnP) specification.

The goals of UPnP are to allow devices to connect seamlessly, and to simplify the implementation of networks at home (data sharing, communications and entertainment) and in corporate environments. This part of the UPnP specification is UPnP-AV (UPnP Audio and Video). The UPnP-AV standards have been referenced in specifications published by other organizations, including Digital Living Network Alliance (DLNA) and others.

The main components of the UPnP-AV (DLNA) network are [1]:

- **MediaServer** is a device that stores and shares digital media, such as music, pictures or movies.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 306–313, 2010.

- **MediaRenderer** is a device that is capable of rendering AV content. Examples of the use of a MediaRenderer include traditional devices such as TVs and stereo systems. Some other contemporary examples include digital devices such as MP3 players or picture frames. However, most of these examples are typically limited to render one specific content type (e.g. a TV typically renders video content).
- **ControlPoint** is a device that is capable of controlling the rendering of digital media streamed from MediaServers to a selected Media Renderer.

The main UPnP-AV (DLNA) components can be combined to form one physical device. For example, pairing a MediaRenderer with a ControlPoint allows one to browse the music content of the available MediaServers and request a MediaServer to start streaming selected files to the MediaRenderer for local playback. There are UPnP-AV (DLNA) components available for most operating systems and many hardware platforms. UPnP-AV (DLNA) devices can either be categorized as software-based or hardware-based. Software-based UPnP-AV (DLNA) devices can run on PCs.

The UPnP network can enable instant access to an entire digital media collection in today's homes. It is important to note that the digital content offered by a MediaServer is not sorted by the physical data structure on the disk, CD, etc., but by the media content itself. For example the music collection can be sorted by Artists, Albums, Genres, Years, etc. The content of such a database can easily exceed 10,000 items. Classical audio or video devices are usually equipped with relatively small displays. If they are acting as a ControlPoint, then selecting one item out of 10,000 can be cumbersome even if the data is well structured.

This is a challenge in which speech recognition can play a very important role. Selecting music or a movie from a very long list by voice is both fast and natural, but using voice control involves certain difficulties, such as automatic extraction of the music metadata, grammar generation, processing music or video metadata in other than the default language, intuitive dialog design, etc. In addition, the interface should be multi-modal, in order to allow standard interaction as well.

In this paper we describe our effort toward implementation of a Voice-enabled UPnP Control Point, which serves as a first step towards reaching these goals. The rest of the paper is organized as follows: In Section 2 we give a brief overview over the content of MediaServer and all the available information relevant for the Speech Recognition. In Section 3 we focus on the processing and usage of available information for Speech Recognition. Section 4 briefly sketches the structure of the ControlPoint, with the focus on multi-modality. Section 5 describes the experiments and Section 6 offers a brief summary.

2 MediaServer Content

As mentioned above, the digital media on the MediaServer is not sorted based on file structure, but is based on the content. To understand the structure we first take a closer look at the possibilities offered by different digital media file types. One particular media file type will be then used to explain the MediaServer data structure.

2.1 Digital Media Content

Digital media data today is stored in a wide range of file formats. Music may be stored in formats like mp3, wma, ogg, etc. Pictures may be stored in formats like jpg, png, etc. Video may be stored in formats like avi, mov, mp4, etc. These file types vary not only in compression or quality level, but also in ability to store additional information related to the digital content.

One of the most popular music formats is MP3. Besides offering good compression and quality, it also offers the possibility of storing metadata, such as title, artist, album, track number and other information about the file content. The most widespread metadata formats are ID3v1 and ID3v2, and the recently introduced APEv2. Important ID3 fields for MediaServer as well as for speech recognition are: *Title, Artist, Album, Year* and *Genre*. As will be demonstrated later, there is no language or text-encoding information that is particularly important for speech recognition. The metadata part of the MP3 file can be created from different sources. It can be taken from the original format, if that is available, obtained from one of the internet music databases like CDDB or FreeDB, it can be added or modified manually, or it can even be left empty. This suggests that metadata extracted from MP3 files is unreliable.

The most popular format for pictures has become JPEG; this is used as a default format for most digital cameras. The *Exif* part of JPEG contains no especially useful information for the MediaServer or for speech recognition. The *Exif* includes technical data about the picture. The only usable information is the time when the picture was created. A similar situation exists with the other graphic formats.

With digital video formats there is a similar situation as there is with JPEG or other picture formats. The most popular video format, AVI, stores some technical information about the video, but other than the time, none of this information is useful for either MediaServer or speech recognition. The recent MPEG-4 offers support for additional information, such as title, chapter name, etc. but it is still rarely used.

2.2 MediaServer Content

The Content Creation Subsystem of the MediaServer extracts the information from the imported data into the internal Content Storage Database (CSD), where it is sorted based on the content obtained from the processed files or from other available sources. The database can then be browsed and searched using a standard UPnP interface and stream the requested content from a MediaServer to the desired MediaRenderer.

MediaServer data is organized as a directory structure of *item*s into a hierarchy of *container*s. Both container and item have several properties, like *id, title, creator*, etc. The top-level root container structure is created automatically based on the content supported by the MediaServer. The title of the top-level containers may differ from one manufacturer to another; this can be true even for MediaServers from the same manufacturer. Figure 1 shows one of the possible top-level container structures.

As seen in (Fig. 1), *Photos* and *Video* containers each have subcontainers. One of these subcontainers — *Directories* was added to this particular MediaServer only recently to satisfy demands for some reasonable content structure of the pictures. Without a directory container the pictures were browsable only by their names (*img_4572.jpg, img_4573.jpg, etc.*), or by their dates.

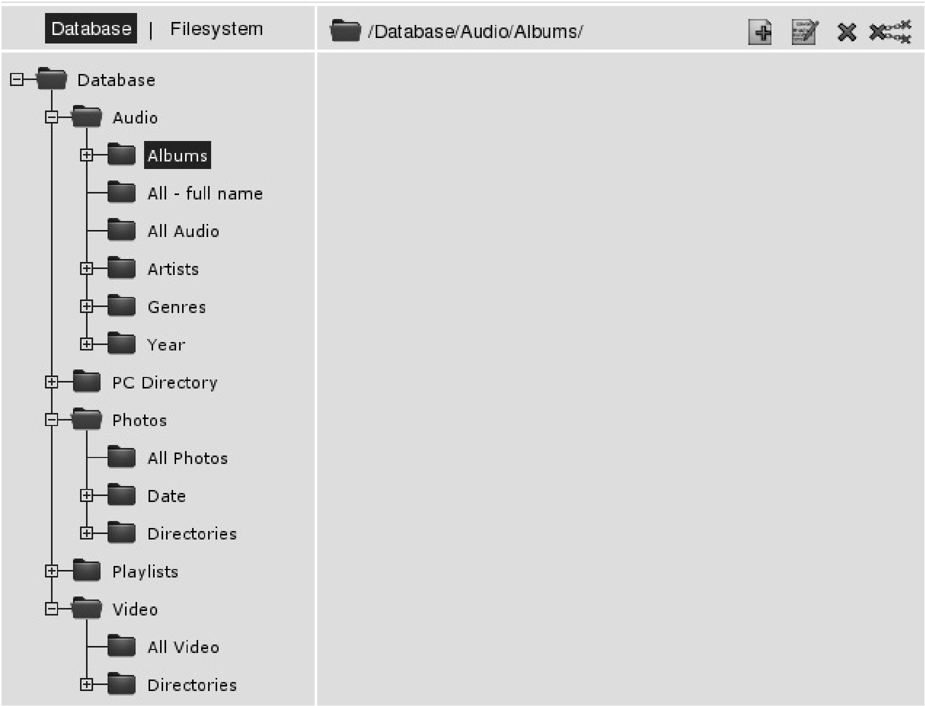


Fig. 1. One of the possible top level container structure

Based on the semantics of metadata in the CSD, containers can provide access to a particular item (media object like song, picture or video) by album title, artist name, etc. The title of the container or item is derived from the content data, if available. Otherwise, the real file name or directory name is used.

The presented MediaServer data structure is browsable or searchable via ControlPoint's GUI[1]. After selecting a particular container or item, the associated URI is sent to the MediaRenderer with the PLAY request.

3 Speech Recognition

As our ControlPoint is intended to run on hand-held devices, we decided to use a grammar-based embedded voice recognizer and allow minimal configurability of the ControlPoint by the user.

We identified the following steps as being of paramount importance in preparing data for the voice recognizer:

- Automatic parsing of the container structure and finding the needed containers.
- Automatic generation of grammars for the recognition system.

We will describe each task below.

[1] Usual interface for UPnP devices.

3.1 Container Structure

First, we need to locate specific containers and, based on names of their items, create lists of albums, songs, pictures, etc. As mentioned above, container structure depends only on the design of the MediaServer. From the mime type[2] of the top level containers it is possible to determine what kind of media data is stored in the container, but this is the only useful information. For example, the path to the container with the “all songs” list for two different MediaServers can be as follows:

- Audio/All Audio
- Music/All Music/Songs

A problem can be caused by a different language of the user interface used by different MediaServers. One MediaServer for different national markets may have the containers' names in different languages. For example, the pictures can be stored in the container *Photos*, and in another case in the container *Bilder*[3].

Manual configuration would not be an option in this case. Setting one MediaServer with music, video and photos would mean having to set at least 5 container paths manually. For more MediaServers in the network the number of set container paths would grow linearly. To automate this task we analyzed the container structure of a few different MediaServers and implemented a simple algorithm to locate the specific container for each requested list (*Album list, Artist list, etc.*). This solution is not robust, as another type of the MediaServer may not be parsed correctly, but UPnP-AV (DLNA) may have been designed for Graphic User Interface only.

3.2 Container Content

The second step is a grammar construction. All grammars addressing dialog issues were created and compiled statically. Grammars for containers and items are generated dynamically, as the content may change when adding or removing media data.

To process multimedia data from an unknown source means dealing with multilingual data without any information about the language as described in 2.1. Multilingualism also brings encoding issues. Considering the fact that the metadata in digital media files can be modified manually also, the detection of the correct code page is an issue in itself. The music collection used for our testing was a compilation of collections provided by several people. It contains music in about 10 languages and 6 encodings. Since we are using a monolingual system for the recognition, it was necessary to filter and modify the input test to pass the pronunciation generator in case the word is not part of the pronunciation dictionary.

The correct pronunciation was the second critical issue in content processing. We are using a combination of a large pronunciation dictionary with manually tuned pronunciation for frequent words and automatic generation. For each unknown word in case of either manual or automatic pronunciation we have to ask three questions:

[2] Part of ProtocolInfo in the container properties.

[3] For German version of a MediaServer.

- What is the language of the unknown word?
- How would the word be pronounced by a native speaker?
- How would the word be pronounced by a non-native speaker who does not speak the language well?

Despite the fact that we are using a monolingual systems (German or English), automatic pronunciation for a well-known artist, or for some common words, also worked well for words from a second language. This was also true in cases of known composers of classical music. For lesser-known names, or for other languages, the automatic pronunciation (especially with the encoding filtering) was not very reliable. For example, a band *Helenine oči* with a simple "*other to ASCII*" filter will result in *Helenine oci*. The German automatic pronunciation system does a good job with this version. The result is something like: [*h e: l n= i: n @ ? o: ts I*]. This version will work for a native German speaker who very likely does not know *Helenine oči*, but can see the band name on the display in the case of a multimodal ControlPoint. Someone who knows the band will pronounce it as [*h E l E J I n E: O tS I*]. To get the pronunciation for the word *oči* is also simple. Because the phonetically Slovak grapheme *č* is the same as German grapheme sequence *tsch*, by replacing each *č* with *tsch*, the Slovak (correct) pronunciation will also be created. However, for the word *Helenine* a simple solution in the case of automatic pronunciation does not exist.

The third issue for the container content is the existence of duplicates. There are three types of duplicates detected so far. The first type is caused by the simple fact that different MediaServers can contain the same data. The second type can be caused by a typo or an encoding issue in the content name. For example, *Herbert Grönemayer* once in UTF-8 and once in ISO-1 will cause this type of duplicates. For the MediaServer these are different artists, and will create two separated containers for them in the Artist list. The third type of duplicates is that of "empty" song name records in the ID3. If no song name has been entered, the usual default is *Track 1, Track 2, etc.*. If there are several such CDs on the MediaServer, then there are several songs *Track 1, Track 2, etc.*.

4 ControlPoint

Using more then one device, including a personal computer, as part of the home entertainment setup is more or less today's standard. This increases the complexity of such systems. In order to minimize the complexity, it is necessary to see the home entertainment system not just as a set of separate devices, but rather as a complex network where users do not need (and do not want) to know the details concerning data storage and distribution, or about the capabilities of each device, etc.

Designed and implemented, the Voice-enabled UPnP Control Point addresses these issues. It simplifies dealing with multiple devices connected to the network of a home entertainment system, and it eliminates the need to configure individual devices. After startup, all devices (MediaServers and Media Renderers) are automatically found. The full scan of the each MediaServer is performed during the next step. The scanned content from different MediaServers is merged together by categories. These categories currently are: *Artist, Album, Song, Photo, Video, Internet Radio*. The user can switch among the categories by voice command, and then select the desired item from the respective category. In case of *Artists, Albums* and *Photos*, the selected item is the container. The content of an entire container is played. For *Video* and *Internet Radio* one selected item is played. For the *Song* the selected song starts to play, and it continues with the next song on the list.

Song selection by voice is also possible within the previously selected container (*Artist, Album*). A set of commands is active independent of the current context like category switching and basic control (*next, previous, stop, random, etc.*). The main category lists are generated and compiled once after the MediaServers scan. The reason for this is a timing issue; the grammars for the required subcontainers are created and compiled on demand.

Beside several MediaServers, several MediaRenderers can be available as well. They usually have different capabilities (can render just certain media types) and can have different locations as well. The MediaRenderers configuration is the only manual procedure required from the user for correct functionality. For each media type the user should define a primary and an alternative renderer. After the *play* command without any target specification the desired content for a particular category is sent to the default renderer for that category. If some additional information is specified, e.g., *Play Keisers Orchestra in the kitchen!*, the alternative renderer is selected. The user may select any renderer via the GUI combo box based on his or her own preferences.

5 Experiments

To test and evaluate the implemented ControlPoint we created a UPnP network with two MediaServers and three MediaRenderers of different capabilities. The content of MediaServers was as follows: 1,014 songs, 114 artists, 98 Albums, 28 picture directories and 57 videos. The title names were in 10 languages and 6 encodings.

The content scanning and grammar compilation, as well as the entire UPnP infrastructure needed for the speech recognition, was running as expected after some tuning. The most interesting aspect (for us) was information about how well speech recognition in such diverse environments would work. For testing purposes, 10 users were playing freely with the system and all their activities, as well as speech data, were logged. The stored logged data were later evaluated. It was possible to use 1,278 recorded utterances for recognition evaluation.

The speech recognition test was performed on the Samsung Q1 portable device with far-field directional microphone. The talking distance was about 40 cm. For speech recognition we used 2 different embedded engines with a 16 kHz German system. The results of both embedded engines were similar. The average sentence error rate was 27.46% (30.58% WER).

We analyzed the results to find the weakness of the system. As expected, the main issue is a title in other than the recognizer's language. The user, especially in case of English titles, is using English pronunciation. For this purpose the use of a bilingual system including English would be beneficial. This is true for today's popular music. In the case of classical or alternative music, a real multilingual system will be more useful.

6 Summary

In this paper we have described various aspects of the development of a voice enabled UPnP-AV (DLNA) ControlPoint. We have explored the UPnP-AV (DLNA) specification and analyzed the possibility to enable voice control of UPnP-AV (DLNA) systems, and have identified available and missing features of the environment for successful speech recognition implementation. Experiments in a real testing environment demonstrated the feasibility of our approach, but also unveiled the need for further research in language detection of written short text and multilingual speech recognition in order to create a system of acceptable accuracy in an environment not initially designed for a voice user interface.

References

1. UPnP specification, http://www.upnp.org
2. Ivanecký, J., Fischer, V., Kunzmann, S.: French–German Bilingual Acoustic Modeling for Embedded Voice Driven Applications. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 234–240. Springer, Heidelberg (2005)
3. Kunzmann, S., Fischer, V., Gonzalez, J., Emam, O., Günther, C., Janke, E.: Multilingual Acoustic Models for Speech Recognition and Synthesis. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Montreal (2004)

Estonian: Some Findings for Modelling Speech Rhythmicity and Perception of Speech Rate

Mari-Liis Kalvik, Meelis Mihkla, Indrek Kiissel, and Indrek Hein

Eesti Keele Instituut, Roosikrantsi 6, Tallinn, Estonia
mariliis@eki.ee, {meelis.mihkla,indrek.kiissel,indrek.hein}@eki.ee
http://www.eki.ee

Abstract. This paper presents the results of two researches with a common aim: to improve the quality of synthetic speech. The study of the parameters of the three quantity degrees which are carrying the Estonian stress structure reveals that the durational ratio of the vowels of stressed and unstressed syllables is the most appropriate distinctive feature of quantity opposition. Investigation of the perception of different speech rates in blind and sighted shows that screenreader trained blinds prefer a considerably higher speech rate.

Keywords: quantity degree, prosody modelling, speech rate.

1 Introduction

Recently the mechanisms of speech rhythmicity have been drawing increasing attention. The focus lies on different aspects of the vowel onset in the stressed syllable, which play a decisive role in enhancing the naturalness of synthetic speech [1]. In Estonian, which is rather a stress-timing language, the foot is where the phonological opposition of three quantity degrees (Q1, Q2, Q3) is realized. The quantity degrees are suprasegmentals [2], whose definition, on the acoustic level, relies on the durational relations of the rime of the stressed syllable and the nucleus of the unstressed syllable, plus the F0 contour [3], making up a complementary system. In addition, the durational relations of adjacent segments and other features have been suggested as important [4]. The present study compares different durational ratios of phonetic quantity and, by statistical modelling, evaluates their significance.

The necessity for the investigation of speech rate perception arose in the course of developing an audio system enabling the blind to listen to reference texts and audio books. By means of the on-line system of the Estonian Library for the Blind [5] the visually impaired can have texts (news, newspapers, magazines and books) read for them and listen to audio books over the Internet. The use of the system revealed that many blind people wish to hear the news and newspaper articles at a considerably higher speech rate than normal. As the system is server-based it cannot afford users tuning the rate smoothly as it would make the system too cumbersome and slow. Hence the need to find some optimal rates to supplement the user menu with two speech rates from the faster-than-normal range, say, fast and very fast.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 314–321, 2010.

2 Recognition and Modelling of Speech Rhythmicity

2.1 Introduction

For quite a while now the problem of the distinctive features of the three quantity degrees – short, long and overlong (Q1, Q2, Q3) – observed in standard Estonian have been subject to discussion among phoneticians. Up to the present the duration ratio between the stressed and unstressed syllable in a foot and, in particular for Q2 and Q3, the difference in their pitch curves have been considered the most important parameters to describe and analyse the quantity degrees of words from differently structured speech, lab-generated as well as spontaneous. As a result of several studies it has been found that the duration ratio between the first and second syllables is 2:3 for Q1, 3:2 for Q2, and 2:1 for Q3 (e.g. [3,6]).

The present study investigates, in addition to the traditional duration ratios presented, the duration ratio of adjacent segments in syllables, and weighs their relevance with statistical methods. As our aim in modelling speech temporal structure is high quality synthetic speech we need the best possible parameters to describe and discriminate between the three Estonian quantity degrees.

This is the first attempt to test the theory of adjacent segments in fluent speech. The theory was developed by A. Eek and E. Meister on the basis of perception tests. It is two stepped and is focused on the ratios of adjacent segments within the main stress syllable and the successive syllable. In a two-syllable word with a vowel-centred structure CV(::)CV, the duration ratio of the vowel (V1) to the consonant (C1) of the first, stressed syllable is supposed to discriminate Q1 words, which have a short V1, from Q2 and Q3 words, whose V1 is long. Q3 words can be distinguished from Q1 and Q2 words by the duration ratio of the vowel (V2) to the inter-vowel consonant (C2) of the second, unstressed syllable. Starting from a belief that perception of the duration difference between two adjacent phones is not possible unless one is 20–25% longer than the other Eek and Meister calculated 1.4 as the limit of duration difference. Words have a short V1 (thus qualifying as Q1) if their V1:C1 ratio is less than 1.4, while for a long V1 (signalling of Q2 or Q3) the ratio needs be equal to or higher than 1.4. A similar ratio computed for the unstressed syllable (V2:C2) supposedly signals of Q3 if it is less than 1.4, while its values equalling or exceeding 1.4 indicate Q1 or Q2, without, however, discriminating between the two [4].

2.2 Material and Method

The material consisted of 485 words (in three quantity degrees), read in sentences by 25 speakers of the Babel linguistic corpus [7], which is based on standard Estonian. Segmentation and phonetic analysis was done using the PRAAT program. Our research is focused on Q1, Q2 and Q3 words with the structure CV(::)CV, most of them disyllabic. In Q1 words the first-syllable vowel is short (e.g. pole [pole] ’is, are not’), whereas in Q2 and Q3 words it is long (e.g. poole [po:le] ’half GenSg’ and poole [po::le] ’towards’, respectively). Thus, the ratio of the stressed and unstressed syllables is found from the ratio of their vowel durations (V1:V2), while the ratio of adjacent phones in the first and second syllables can be written as V1:C1 and V2:C2, respectively. A small number of the words begin either with a vowel or with a consonant cluster. The analysed material contains stressed as well as unstressed words from different positions (initial, middle, final) in the sentence or phrase.

2.3 Results

The results of sound measurement and the ratios computed have been summarized in Table 1. The material contained 234 Q1 words, 150 Q2 words and 101 Q3 words.

Table 1. Mean durations (ms) and duration ratios of the first-syllable consonant (C1) and vowel (V1), inter-vowel consonant (C2) and vowel of the second syllable (V2) in Q1, Q2, Q3 words

	C1	**V1**	**V1:C1**	**C2**	**V2**	**V1:V2**	**V2:C2**
Q1	69	68	**1.05**	52	87	0.82	**1.74**
Q2	64	120	**1.99**	52	69	1.80	**1.37**
Q3	66	165	**2.59**	59	66	2.59	**1.17**

First, V1:V2 is the classical ratio to be examined. The mean durations easily reveal that in Q1 words V1 is about twice as short as in Q2 and Q3 words, which is generally considered sufficient to perceive the short/long opposition between Q1 and the rest (the short Q1 vs. the long Q2 and overlong Q3). Comparing our results with those received on laboratory speech earlier we find that our ratios are realistic. A typical duration ratio of Q1 falls in the interval 0.6–0.7 ([6,8,9,10]), but it can also range up to 1.0. For Q2 the ratios of V1:V2 vary more across different studies, ranging from 1.2–1.60; for Q3 the range is 2.4–2.6 [8,10,11]. The CV(::)CV structure has also been a research subject for E. L. Asu et al. [12], who study spontaneous speech. According to their results the average duration ratio is about 0.7 for Q1, 1.7 for Q2 and over 2.0 for Q3. Those figures do not contradict ours either.

Secondly we consider the ratios of adjacent phones for stressed vs. unstressed syllables. Our results for a stressed syllable confirm that V1 is short (as expected for Q1) if and only if V1:C1 is less than 1.4 (1.05 in Table 1). For Q2 and Q3 words the respective ratios are 1.99 and 2.59, which are both higher than 1.4. In an unstressed syllable the ratio of the vowel V2 and its preceding consonant C2 is perhaps a little less unambiguous. Still, the theoretical ratio for Q1 and Q2 words being 1.4 or higher, our result for Q1 (1.8) should do well, and so does the ratio for Q2 as it can be rounded to 1.4 easily. Also, our V2-to-C2 ratio for Q3 words supports the theory as 1.17 is clearly lower than 1.4. Thus, our material indeed seems to corroborate Eek's theory.

Next, statistical methods will be used to find out which duration ratios, the traditional V1:V2 or the two-step system of V1:C1 → V2:C2 suggested by Eek, are vital for classifying quantity degrees. Figure 1 presents a CART-generated decision tree. In this tree the quantity degree correlates strongly with the V1:V2 ratio, whereas the other features proved to be insignificant for this model.

From Figure 1 we can see that for Q1 the duration ratio V1:V2 is less than 1.25, while for Q2 the values of the ratio range from 1.25–1.98 and the criterion for recognizing Q3 is a V1-to-V2 ratio that exceeds 1.98. Thus the material is in general divided

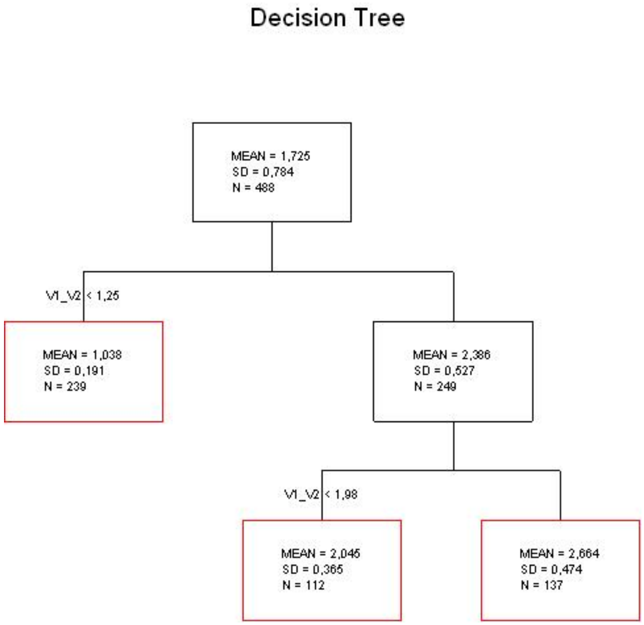


Fig. 1. Decision tree based on the classical duration ratio of V1:V2

into three groups, similarly to the classical division into the three quantity degrees. A similar decision tree based on two duration ratios (V1:C1 and V2:C2) actually manages to classify the quantity degrees by using only one of them (V1:C1) as the sufficient criterion. This points to the length (short or long) of the first syllable vowel V1 and thus, to the primary (durational) division of the quantity degrees into short (Q1) and long ones (Q2, Q3). Consequently, for our material the V2:C2 criterion has proved marginal after all. This brings back memories of M. Hint's [13] theory of syllabic quantity degree, arguing that although phonetic quantity is manifested in the foot, its degree depends on certain parameters of the stressed syllable. For weighing the relevance of duration ratios in the model of phonetic quantity simple equations of linear regression were generated. For classical duration ratios the linear model yielded quite a strong correlation between the input and output (correlation coefficient r=0.867). Consequently the model explains over 75% of the data variation (coefficient of determination r^2=0.752). The alternative model generated from the other two duration ratios, yielded a correlation coefficient equal to 0.759, which means that it explains only 58% of the variation in the data analysed. According to the above results, the classical duration ratio (V1:V2) is the most relevant parameter to be considered in modelling the temporal structure of Estonian speech. However, although the above parameter guarantees close correlation between the model input and output, similar tests should be run for other features, e.g. intensity and pitch, to find out their possible role in the formation of phonological oppositions.

3 Speech Rate Perception

Although an on-line audio system is meant for the visually impaired mainly, our test of speech rate perception was also applied to a group of sighted subjects, for comparison. This was meant to answer such questions as: What speech rates are preferred by the blind vs. the sighted? Is the preference of very fast speech rates by the visually impaired a myth or not? Is there such a thing as an optimal speech rate?

3.1 Subjects

The test was taken by 58 blind or visually impaired subjects (29 female and 29 male, aged 14–79) and by 56 sighted subjects (41 female and 15 male, aged 18–58). For all subjects, Estonian was the mother tongue.

3.2 Test Material

The stimuli for the test of speech rate perception were generated from two audio books ("American tragedy" by T. Dreiser, male voice, and "Das erste Mal und mehr" by E. Stein-Fischer, female voice, both in Estonian translation) and some news fragments, synthetic voice. The latter was produced by a diphone-based Estonian text-to-speech synthesizer [14], using an MBROLA synthesis motor. The synthetic voice was generated in two variants, one using a rule-based prosody model (SYNT1), the other a statistical one (SYNT2). This was to test the impression of the blind that in the case of a synthesizer with a statistical prosody module, rate quickening would lower the quality of output speech. For the female voice the natural reading rate was 135 words per minute, the male voice making 122 words per minute; the synthesizers were tuned to match the female rate. For each voice, eight speech samples of 35–55 sec were generated, each of a different speech rate (see Table 2).

Table 2. Speech samples as stimuli of different speech rates (natural speech rate 100% = 135 s/min)

81	108	**135**	162	189	216	243	270	**words/min**
60	80	**100**	120	140	160	180	200	%

The rate of the samples from audio books was regulated using the signal processing program Adobe Audition 3 for high precision time compression with time stretch (which preserves pitch). The subjects were exposed to the speech rate stimuli by voice series presented in a random order. The appropriateness of the speech rate was asked to be evaluated in a five-point system (5 – the best, 4 – good, 3 – tolerable, 2 – uncomfortable, 1 – unsuitable, i.e. unintelligible, too quick or too slow).

3.3 Results

Figure 2 presents the average blind vs. sighted scores for different speech rates. The left diagram shows the scores given for the female voice and the synthetic voice SYNT1, the right one for the male voice and the synthetic voice SYNT2. According to the graphs the blind prefer the speech rates 1.2 and 1.4, which are 20% and 40% faster than natural, respectively. The sighted appreciate natural speed highly, but 1.2 and 1.4 are not considered bad either.

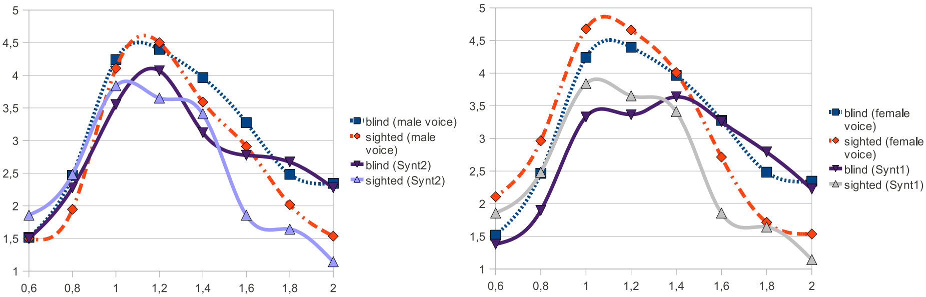


Fig. 2. The average scores from the blind and the sighted for the human and the synthetic voices at different speech rates

For a male voice 1.2 was considered the best speech rate both by the sighted and the blind. The results are probably due to the natural speech rate for a male voice, which is about 10% slower than the rest (122 s/min vs. 135 s/min).

Although the subjects were asked to evaluate the suitability of the speech rate only, not the pleasantness of the voice, synthetic speech scored almost a point lower, on average, than human speech. However, the prosody module of the synthesizer (SYNT1 vs. SYNT2) does not seem to have influenced the score significantly at faster speech rates. Thus the results fail to support the idea that the quality of the SYNT2 voice might deteriorate at quicker rates.

The test proved that the ratings of the blind and the sighted differ far less than first believed. Figure 3 presents the average scores from all visually impaired subjects. However, the blind include many who seldom use a computer, if at all, and who thus lack the experience of listening to synthetic speech at different speech rates. Figure 3 presents the average scores given to different speech rates vs. the previous experience, in years, of the visually impaired subjects with a computer and screen reader. The results reveal an obvious tendency that in the visually impaired, longer practical “training”, i.e. experience of using the above devices causes an increase in the ability of understanding rapid speech.

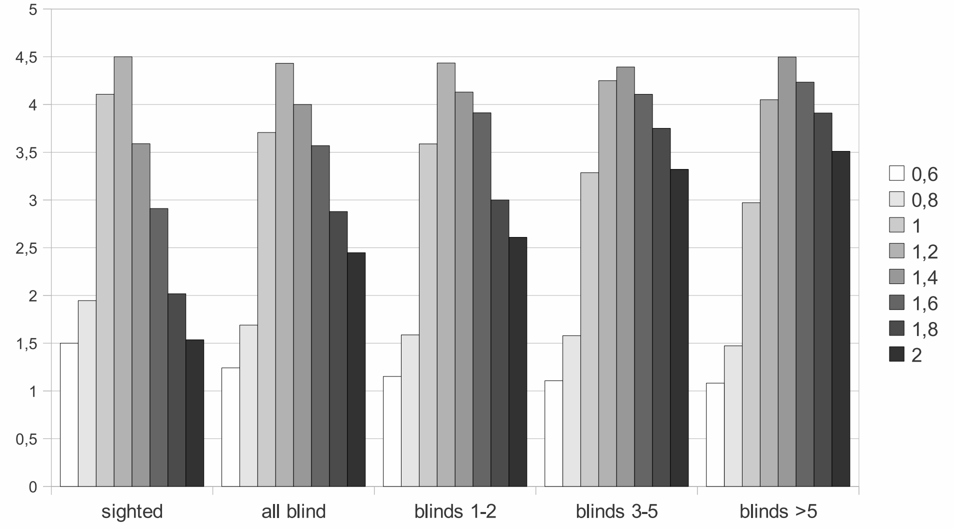


Fig. 3. The average scores given to different speech rates vs. the previous experience, in years, of the visually impaired subjects with a computer and screen reader

4 Conclusions

The aim of the study was to find out whether Estonian quantity degrees could be distinguished by any other ratio but the traditional duration ratio of V1:V2. Our analysis of copious data proved that adjacent sound ratios are not as relevant as the ratio of the first and second syllable sounds. When modelling speech temporal structure one should keep in mind that standard Estonian is characterized by an alternation of words of three different quantity degrees, based on a natural alternation of stressed and unstressed syllables. The quantity degrees can be distinguished by a comparison of the duration ratios of those syllables, but obviously this is not all there is to it. The next object of research relevant in this respect should be the manifestation and role of pitch. The conducted test of speech rate perception did not provide an unambiguous answer to all set questions. There are, indeed, certain differences observable between the speech rates preferred by the blind and the sighted, but the level of the difference depends not on the visual impairment but rather on the subjects' experience with using a computer and a screen reader. The ability to understand rapid speech appears after about three years of everyday practice of using of a computer and listening to synthetic speech. Considering the results of the tests the audio system was supplemented, in addition to the normal speech rate, with a fast rate (1.3 times, i.e. 30% faster than normal) and a very fast rate (1.6 times, i.e. 60% faster than normal) for an advanced computer user.

Acknowledgments

This work has been supported by the National Programme for Estonian Language Technology, grant ETF7998 and project SF0050023s09.

References

1. Keller, E., Port, R.: Speech Timing: Approaches to Speech rhythm. In: Trouvain, J., Barry, W.J. (eds.) Proceedings of the 16th International Congress of Phonetic Sciences. Saarbrücken, Saarbrücken, August 6-10, pp. 327–329 (2007)
2. Ross, J., Lehiste, I.: The Temporal Structure of Estonian Runic Songs. In: Lahiri, A. (ed.) Phonology and Phonetics, vol. 1. Mouton de Gruyter, Berlin (2001)
3. Lehiste, I.: Search for Phonetic Correlates in Estonian Prosody. In: Lehiste, I., Ross, J. (eds.) Estonian Prosody: Papers from a Symposium, pp. 11–35. Institute of Estonian Language, Tallinn (1997)
4. Eek, A., Meister, E.: Foneetilisi katseid kvantiteedi alalt. Keel ja Kirjandus (11–12), 815–837, 902–916 (2003)
5. Estonian Library for the Blind, http://www.epr.ee/kalev
6. Lehiste, I.: Segmental and Syllabic Quantity in Estonian. In: American Studies in Uralic Linguistics, Bloomington, Indiana University, vol. 1, pp. 21–28 (1960)
7. Eek, A., Meister, E.: Estonian Speech in the BABEL Multi-Language Database: Phonetic-phonological problems revealed in the text corpus. In: Fujimura, O. (ed.) Proceedings of LP 1998, vol. II, pp. 529–546. Karolinum Press, Prague (1999)
8. Liiv, G.: Eesti keele kolme vältusastme kestus ja meloodiatüübid. Keel ja Kirjandus (7–8), 412–424, 480–490 (1961)
9. Eek, A.: Kvantiteet ja rõhk eesti keeles (I). Fonoloogiliste tõlgenduste kriitikat. Keel ja Kirjandus 9, 481–489 (1983)
10. Eek, A., Meister, E.: Simple Perception Experiments on Estonian Word Prosody: Foot Structure vs. Segmental Quantity. In: Lehiste, I., Ross, J. (eds.) Estonian Prosody: Papers from a Symposium, pp. 77–99. Institute of Estonian Language, Tallinn (1997)
11. Krull, D.: Stability in Some Estonian Duration Relations. In: Experiments in Speech Processes. PERILUS (Phonetic Experimental Research, Institute of Linguistics, University of Stockholm), vol. (XIII), pp. 57–60 (1991)
12. Asu, E.L., Lippus, P., Teras, P., Tuisk, T.: The Realization of Estonian Quantity Characteristics in Spontaneous Speech. In: Aaltonen, O., Aulanko, R., Vainio, M. (eds.) Nordic Prosody – Proceedings of the Xth Conference, Helsinki 2008, pp. 49–56. Peter Lang Verlag, Frankfurt (2009)
13. Hint, M.: Prosoodiaväitlustes läbimurdeta. Keel ja Kirjandus (3–5), 164–172, 252–258, 324–335 (2001)
14. Mihkla, M., Meister, E.: Eesti keele tekst-kõne-süntees. Keel ja Kirjandus 45(2-3), 88–97, 173–182 (2002)

Using Gradient Descent Optimization for Acoustics Training from Heterogeneous Data

Martin Karafiát*, Igor Szöke, and Jan Černocký

Brno University of Technology, Faculty of Information Technology
Department of Computer Graphics and Multimedia, Speech@FIT
Božetěchova 2, Brno, Czech Republic
{karafiat,szoke,cernocky}@fit.vutbr.cz

Abstract. In this paper, we study the use of heterogeneous data for training of acoustic models. In initial experiments, a significant drop of accuracy has been observed on in-domain test set if the data was added without any regularization. A solution is proposed by getting control over the training data by optimization of the weights of different data-sets. The final models shows good performance on all various tests linked to various speaking styles. Furthermore, we used this approach to increase the performance over just the main test set. We obtained 0.3% absolute improvement on basic system and 0.4% on HLDA system although the size of the heterogeneous data set was quite small.

1 Introduction

The amount of in-domain training data has significant effect on the accuracy of speech-to-text transcription systems based on Hidden Markov Models (HMM). Speaking style is often the major cause of variability in the data, therefore only in-domain data are typically used for HMM training. Our target domain is recognition of spontaneous Czech continuous telephone speech (CTS), but good performance on non-spontaneous data (radio) is also desired. In our initial experiments, we observed significant drop off accuracy if models were trained just on CTS and tested on non-spontaneous data.

An intuitive solution is to add non-spontaneous speech to the training process. This is however not trivial as it caused significant drop of accuracy on in-domain (CTS) test set. Weighting of training data is necessary, but doing this manually is difficult especially if the amount of different training data sources is high. Therefore, we defined an objective function, and perform HMM training with respect to minimization of this function. Weighting of training data set could be done by removing some speech segments or better, by weighting of statistics needed for HMM estimation.

1.1 Weighting of HMM Statistics

Forward-backward algorithm is common algorithm used for estimation of HMMs. It generates occupation probabilities $\gamma_j(t)$ for Gaussian mixture component j, which allow to gather sufficient statistics for re-estimation formulae:

* This work was partly supported by Czech Ministry of Interior project No. VD20072010B16, Grant Agency of Czech Republic project No. 102/08/0707, Czech Ministry of Education project No. MSM0021630528, and by BUT FIT grant No. FIT-10-S-2.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 322–329, 2010.

$$\gamma_j = \sum_{t=1}^{T} \gamma_j(t) \tag{1}$$

$$\boldsymbol{\theta}_j(\mathbf{O}) = \sum_{t=1}^{T} \gamma_j(t)\mathbf{o}) \tag{2}$$

$$\boldsymbol{\theta}_j(\mathbf{O}^2) = \sum_{t=1}^{T} \gamma_j(t)(\mathbf{o}(t))^2 \tag{3}$$

Then, Gaussian parameters are re-estimated according to:

$$\hat{\boldsymbol{\mu}}_j = \frac{\boldsymbol{\theta}_j(\mathbf{O})}{\gamma_j} \tag{4}$$

$$\hat{\boldsymbol{\sigma}}_j = \frac{\boldsymbol{\theta}_j(\mathbf{O}^2)}{\gamma_j} - \boldsymbol{\mu}_j^2 \tag{5}$$

In case of training from various databases **D**, it is possible to divide the collection of statistics into database-specific parts:

$$\gamma_j = \sum_{d=1}^{D} w(d)\gamma_j^d, \tag{6}$$

$$\boldsymbol{\theta}_j(\mathbf{O}) = \sum_{d=1}^{D} w(d)\boldsymbol{\theta}_j^d(\mathbf{O}), \tag{7}$$

$$\boldsymbol{\theta}_j(\mathbf{O}^2) = \sum_{d=1}^{D} w(d)\boldsymbol{\theta}_j^d(\mathbf{O}^2), \tag{8}$$

where $w(d)$ is weight of database d. When starting, all values $d(w)$ are initialized to one. Next, it is possibly to enhance system performance by optimization of objective function $F(M(\mathbf{w}))$, which includes accuracies on different test-sets. It depends on the current model set $M(\mathbf{w})$, where $\mathbf{w}$ is the vector of all weights $w(d)$.

2 System Optimization

2.1 Gradient Descent Approach

Gradient Descent (GD) optimization is a general algorithm that can be directly used for our purpose. It uses derivatives of the objective function to find the steepest gradient. In the process, the optimized variables are moved in negative direction which will reduce the value of the function.

The process is iterative and can be described as follows:

1. Compute the derivative $dF(x)/dx$ of the function $F(x)$ with respect to its independent variables x.

Table 1. Language model training data

Corpora	amount of words [W]
CTS training data	685k
PMKBMK	1182k
subtitles	192M

2. Change the value of x according to

$$x^{(n+1)} = x^{(n)} - \eta \frac{dF(x^{(n)})}{dx}, \tag{9}$$

   where η is learning rate.
3. Repeat above steps till convergence is reached.

The learning rate η is a control value which significantly influences the convergence of the algorithm. If it is too big, a step will overshoot the minimum of $F(x)$. If too small, the process will need long time to converge. Therefore, we used variable learning rate:

$$\eta^{(n)} = \begin{cases} 1.2\eta^{(n-1)} \text{ if } \ F(x) \text{ increases} \\ 0.5\eta^{(n-1)} \text{ if } F(x) \text{ decreases,} \end{cases} \tag{10}$$

which guarantees fast convergence while not overshooting the minimum.

3 Experimental Setup

The speech recognition system is based on HMM cross-word tied-states triphones. MF-PLP features were generated using HTK, with a total number of 13 coefficients. Deltas, double-deltas and (in the HLDA system), triple-deltas were added, so that the feature vector had 39 or 52 dimensions respectively. Cepstral mean and variance normalization was applied with the mean and variance vectors estimated on speaker basis. VTLN warping factors were applied by adjusting the centers of the Mel-filters. HLDA was estimated with Gaussian components as classes and the dimensionality was reduced to 39.

All tests in this paper used 2-gram language models trained on corpora described in Table 1. PMKBMK is Prague and Brno corpus of spoken Czech[1]. The corpus of subtitles was obtained from the web. The size of dictionary was 1M words[2].

3.1 Data Description

The training and test data was collected from various sources which significantly differ in channel and speaking style:

[1] http://korpus.cz/english/pmk.php, http://korpus.cz/english/bmk.php

[2] Note, that Czech is highly inflective language and 50k dictionary usual in English systems is far too small.

Table 2. Training and test data

Training set	amount of data [h]	Test set	amount of data [h]
train-C	46	testCTS	2.2
train-R	19	testR	1
train-L	21	testL	0.5
train-P	15	testP	0.5
train-V	2.2	-	

Table 3. Effect of adding databases into the training

Models	testCTS	testR	testP	testL
train-C	52.6	54.2	39.5	37.2
train-CR	52.3	63.2	49.3	46.5
train-CP	52.2	58.1	62.4	53.6
train-CL	51.7	59.2	57.3	56.7
train-CV	52.6	56.3	41.4	39.8
train-CRPLV	51.3	64.2	63.0	59.1

- CTS - “train-C” - Spontaneous telephone speech.
- RadioCTS - “train-R” - People calling to radio during broadcasts. Partly spontaneous speech.
- Liberec - “train-L” - Read speech (over the telephone) from University of Liberec.
- Plzen - “train-P” - Read speech (over the telephone) from University of South Bohemia.
- 158 - “train-V” - Emergency calls. Very short spontaneous recordings.

The amounts of data taken for training and test can be found in Table 2.

4 Experiments

As spontaneous telephone speech was our main domain, the initial model set was trained on CTS data only. Next, we investigated the effect of adding each particular database into the training process. The initial models were re-trained by additional iterations of standard maximum likelihood (ML) training. The results are given in Table 3.

We observed that adding any data into the training does not improve and even mostly degrades the performance on “testCTS”. On contrary, initial CTS models have quite poor performance on other tests – a system based on these models will work poorly in case it has to deal with read data instead of spontaneous (which can easily happen in a real application). Using all corpora, without any weighting, significantly improves the performance on all non-CTS tests but causes 1.3% drop on “testCTS” – this is not satisfactory as CTS is our main domain.

To verify, that this misbehavior was caused by different speaking styles, we run same experiment with speaker-based adaptation for testing. This adaptation performs more

Table 4. Effect of adding databases into the training with application of CMLLR

Models	testCTS	testR	testP	testL
train-C	54.6	62.2	46.0	47.6
train-CR	54.6	66.9	56.1	55.3
train-CP	54.1	65.0	64.6	59.3
train-CL	54.3	65.4	61.4	61.7
train-CV	54.7	63.0	48.0	48.2
train-CRPLV	53.8	67.4	65.2	63.6

efficient speaker and channel compensation than basic CMN/CVN used in experiments above. Constrained Maximum Likelihood Linear Regression (CMLLR) was taken for this purpose [1,2]. The results can be found in Table 4. We observed generally better results than in the experiment without adaptation, but the same trends with different training data.

4.1 Gradient Descent Training

It is obvious that system degradation on "testCTS" caused by adding non-CTS data can be reduced by minimizing their influence. It could be simply done by setting up the weighting coefficients to values smaller than one, see equations (6)–(8).

Optimal weights could also be found by exhaustively running many tests, or automatically by optimization of objective function. We defined the objective function $F(\mathbf{w})$ as a weighted sum of word error rates (WER) of the system for each test set. The weights were set according to performances expected on particular test sets.

For our application, we wanted to minimize the accuracy drop on "testCTS" and put smaller importance on other tests, so that the objective function was defined as:

$$\begin{aligned} F(\mathbf{w}) = 0.6A_{testCTS}(M(\mathbf{w})) + 0.2A_{testR}(M(\mathbf{w})) + \\ +0.1A_{testP}(M(\mathbf{w})) + 0.1A_{testL}(M(\mathbf{w})), \end{aligned} \tag{11}$$

where $A_x(M(\mathbf{w}))$ is WER of the resulting model M on test set x, and $\mathbf{w}$ is vector of weights used for merging of statistics in equations (6)–(8). Obviously, the coefficients multiplying accuracies could be set differently, depending on the target application.

Training acoustic models with respect to optimization of $F(\mathbf{w})$ can be described in the following steps:

1. Initialize weights. The most simple initialization is:

$$\mathbf{w} = \left[\ 1.0\ 1.0\ 1.0\ 1.0\ 1.0\ \right],$$

   where columns represent weights for training databases ["C" "R" "P" "L" "V"]. Train initial models and evaluate $F(\mathbf{w})$.
2. Collect statistics for all training sets.
3. Estimate $dF(\mathbf{w})/d\mathbf{w}$ where $d\mathbf{w}$ is approximated by a small step. This is quite painful operation due to the need to run decoding across all test sets for each

$$\left[\frac{dF(\mathbf{w})}{dw_1} \quad \cdots \quad \frac{dF(\mathbf{w})}{dw_D} \right]$$

Table 5. Gradient Descent optimization with and without CMLLR

Optimization tests run without adaptation				
Models	testCTS	testR	testP	testL
train-C	52.6	54.2	39.5	37.2
train-CRPLV	51.3	64.2	63.0	59.1
GD optim train-CRPLV	52.1	64.1	61.7	58.3
w = [1.21 0.92 0.60 0.66 1.60]				
Optimization tests run with CMLLR				
train-C	54.6	62.2	46.0	47.6
train-CRPLV	53.8	67.4	65.2	63.6
GD optim train-CRPLV	54.1	67.0	64.8	62.7
w = [1.23 0.92 0.95 0.80 1.09]				

4. Update new weights according to equation (9).
5. Normalize weights to sum to the same number as initial weights.
6. Estimate new models set, evaluate $F(\mathbf{w})$ and go back to step 3. These iterations are run until the process stabilizes.
7. Go back to step 2 and iterate until the whole process stabilizes.

Note, that returning to step 2 can change the optimal weights. Therefore, we fix the number of gradient descent iterations to a fixed value, and stop the training when the whole process, including ML iterations, is stabilized.

The final results are shown in Table 5. Degradation of accuracy is now 0.5% instead of 1.3% absolute and the models perform well also for other data.

Using optimized weights in HLDA systems. The Heteroscedastic Linear Discriminant Analysis (HLDA) [3] can be used to derive a linear projection de-correlating feature vectors and reducing their dimensionality. It is nowadays a common technique used for enhancing speech recognition systems. For implementation into this scheme, we investigated two possibilities, both used fixed weights from the optimization of the basic system:

- HLDA is estimated with the initial set of models – training using all data is done by additional ML iterations with fixed GD weights. It is denoted “GD train-C HLDA + RPLV” in Table 6.
- Models coming from the optimization above are used to estimate HLDA statistics from all the data. Statistics are merged with fixed weights, new HLDA transformation matrix is estimated and new models are trained. This is denoted “GD train-CRPLV HLDA” in Table 6.

Table 6 shows smaller drop of accuracy on “test-C” with fixed HLDA than with the retrained HLDA. This is caused by using just close-domain data for HLDA estimation.

Optimization for testCTS only. In initial experiments, we have found that non-spontaneous data has no positive effect in training of CTS system if the weights are set

Table 6. Using GD weights in HLDA system

Models	testCTS	testR	testP	testL
train-C	52.6	54.2	39.5	37.2
train-C HLDA	53.5	56.7	41.8	42.9
GD train-C HLDA + RPLV	53.2	65.9	63.8	60.4
GD train-CRPLV HLDA	52.7	66.2	64.2	61.2

Table 7. Gradient descent optimization for enhancing accuracy just on "testCTS" set

Models	testCTS
train-C CMLLR	54.6
train-C-V CMLLR	54.7
GD optim train-CRPLV CMLLR	54.9
w = [3.15 0.32 0.14 0.03 1.36]	

Table 8. Gradient descent optimization for enhancing accuracy just on "testCTS" set with using HLDA

Models	testCTS
train-C HLDA CMLLR	56.1
GD train-C HLDA + RPLV CMLLR	56.1
GD train-CRPLV HLDA CMLLR	56.5

to be equal. Therefore, we run an experiment where we changed the objective function just to accuracy on "testCTS":

$$F(\mathbf{w}) = 1.0 A_{testCTS}(M(\mathbf{w})) \tag{12}$$

This is similar to an approach often used in language modeling – use out-of-domain corpora for enhancing the performance on current task [4].

The initial weights were set close to the expected final values, only "train-C" and "train-V" weights were set to non-zero values, as they show positive effect on accuracy:

$$\mathbf{w} = \left[\ 3.0\ 0.0\ 0.0\ 0.0\ 2.0\ \right],$$

The whole optimizations run with CMLLR in the tests. Table 7 presents 0.2% absolute gain from models trained on improving data only ("train-CV") and 0.3% against "train-C" baseline. On the resulting weights, we see that main importance was put on in-domain data and partly on "train-V" and "train-R" training sets. The read speech corpora contribute almost zero.

In experiments in Section 4.1, we have shown no positive effect on "testCTS" from retraining HLDA on all data if weights are optimized to produce balanced model for all tests. But if weights are optimized for this test, we can benefit from re-estimation of HLDA transformation on all data – Table 8 shows 0.4% absolute improvement.

5 Conclusions

In this paper, we have studied the use of heterogeneous data for the training of acoustic models. To obtain desired performance, we used regularization based on optimization over the data weights. The regularization was found to be very useful to increase robustness of acoustic models to various speaking styles, or for use of heterogeneous data for a single target application. In case of enhancing the training data, we obtained 0.3% absolute improvement for the basic system and 0.4% for the HLDA one. Note, that the amount of heterogeneous data was smaller than for in-domain data. It is therefore realistic to expect even more improvement with increased size of out of domain data data. In our future work, we will investigate the possibility to run the optimization together with Speaker Adaptive Training [5,2].

References

1. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
2. Gales, M.: Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition (1997)
3. Kumar, N.: Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition. Ph.D. thesis, John Hopkins University, Baltimore (1997)
4. Iyer, R., Ostendorf, M., Gish, H.: Using Out-of-Domain Data to Improve In-Domain Language Models. IEEE Signal Processing Letters 4(8), 221–223 (1997)
5. Tsakalidis, S., Byrne, W.: Acoustic Training from Heterogeneous Data Sources: Experiments in Mandarin Conversational Telephone Speech Transcription. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005), March 18-23, vol. 1, pp. 461–464 (2005)

Recovery of Rare Words in Lecture Speech*

Stefan Kombrink[1], Mirko Hannemann[1], Lukáš Burget[1], and Hynek Heřmanský[1,2]

[1] Speech@FIT, Brno University of Technology, Czech Republic
[2] Johns Hopkins University, Baltimore, USA
{kombrink,ihannema,burget,iszoke}@fit.vutbr.cz, hynek@jhu.edu

Abstract. The vocabulary used in speech usually consists of two types of words: a limited set of common words, shared across multiple documents, and a virtually unlimited set of rare words, each of which might appear a few times only in particular documents. In most documents, however, these rare words are not seen at all. The first type of words is typically included in the language model of an automatic speech recognizer (ASR) and is thus widely referred to as in-vocabulary (IV). Words of the second type are missing in the language model and thus are called out-of-vocabulary (OOV). However, these words usually carry important information.

We use a hybrid word/sub-word recognizer to detect OOV words occurring in English talks and describe them as sequences of sub-words. We detected about one third of all OOV words, and were able to recover the correct spelling for 26.2% of all detections by using a phoneme-to-grapheme (P2G) conversion trained on the recognition dictionary. By omitting detections corresponding to recovered IV words, we were able to increase the precision of the OOV detection substantially.

1 Introduction

Since the early days of speech recognition, systems have been limited to vocabularies, which cover just the most common words. In many applications, however, the vocabulary is potentially unlimited. To deal with an unlimited vocabulary, one could either use techniques such as confidence measures [1,2], which have been developed to detect misrecognized speech due to OOV content. Another way is to use open vocabulary speech recognition systems, which enhance their vocabulary by using sub-words as e.g. in [3] and [4].

In this work, we also used a hybrid word/sub-word recognizer, where we modeled words and sub-words hierarchically as described in [5]. The word/sub-word recognizer detects OOV words, and retrieves their pronunciations as a preliminary description. Based on that, we used graphone models [6] to convert the preliminary phonetic description into a corresponding word spelling, what we refer to as *recovery of rare words*.

The focus of this work is on the recovery of rare IV and OOV words rather than on pure OOV detection. The spelling of detected sub-word sequences might be of interest due to several reasons:

* This work was partly supported by European project DIRAC (FP6-027787), by Grant Agency of Czech Republic project No. 102/08/0707, Czech Ministry of Education project No. MSM0021630528 and by BUT FIT grant No. FIT-10-S-2.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 330–337, 2010.

- **correction:** We could substitute the sub-word sequences by the estimated spelling. By retrieving the estimated spellings of OOV words and even IV words, which got misrecognized as sub-word sequences for various reasons, we are able to correct a significant portion of them.
- **interpretability:** Non-linguists are not necessarily familiar with phonetic alphabets and find it bothersome to interpret such a description. However, they are able to guess a pronunciation from an approximate word spelling, even if seen for the first time and without knowing the meaning of the word.
- **consistency:** P2G conversion techniques are supposed to model the phonological relationship between pronunciations and spellings of words within a language. Especially if we train our P2G model on the dictionary used in the recognizer, the estimated spelling is consistent with existing words in the dictionary[1].

2 Hybrid Word/Sub-word Recognition

The utilized hybrid recognizer can be seen as a combination of two differently constrained recognizers interacting in parallel. The strongly constrained part takes into account word context by using a vocabulary of most frequent words and a general-purpose OOV word. The weakly constrained part models rare words using sub-word sequences by using a set of most frequent sub-word units. In the search for the most likely output sequence given an acoustic observation, the hybrid recognizer decides for the best path based on the overall likelihood of the composite word/sub-word model.

In the following, we show an example comparing the output of a standard word recognizer and our hybrid word/sub-word recognizer.

```
 REFERENCE: ...BACK TO BELGIUM(OOV) </s>
  WORD REC: ...BACK TO BALANCE THEM </s>
HYBRID REC: ...BACK TO <unk> b_ae_l jh_ih_ah_m <phnsilsp> </s>
```

The special words <unk> and <phnsilsp> mark transitions between the strongly and the weakly constrained parts of the recognizer. No word with an appropriate pronunciation for “BELGIUM” was found in the recognition dictionaries. The hybrid recognizer, however, was given the freedom to describe the missing word as a sequence of sub-word units, and hereby retrieving implicitly a phonetic description.

3 Partial OOV Detection

Generally speaking, the time region and the pronunciation estimated by the hybrid recognizer do not always match the reference precisely, e.g.:

```
  REFERENCE: ...GOES SUPERSTRING(OOV) THEORY WHAT...
RECOGNITION: ...GOES TO <unk> p_r_ih s_t_r_ih_ng p_iy r_iy <phnsilsp> WHAT...
```

[1] E.g. s_eh_n_t_axr recovers to CENTRE when trained on British English vs. CENTER when trained on American English.

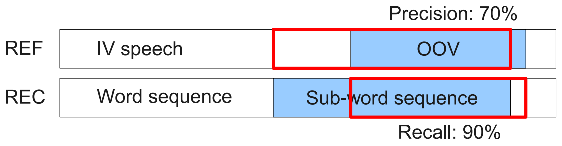


Fig. 1. Definition of Precision and Recall of a partially detected OOV Word: Precision is the percentage of OOV speech contained in the actual sub-word sequence, whereas recall is the percentage of OOV speech being detected

Hence, we decided to call sub-word sequences *partial OOV detections*. In the scope of this work, we are mainly interested in those partial detections which cover the time regions of the reference OOV word to an extent, that allows to recover the spelling from the retrieved phonetic transcription. Thus, based on the overlap between the boundaries in the reference and the recognition, we defined, how well a detected region will be suited for a word recovery task.

Figure 1 shows a typical partial OOV detection with inaccurate boundaries. The reference OOV word is being compared to the recognition of our hybrid word/sub-word decoder. Consequently, the sub-word sequence is interpreted as detected OOV, and the quality of that partial detection can be expressed precision and recall. We used the symmetric f-score to express the quality of a detected OOV word by just a single number:

$$f = \frac{2 \times precision \times recall}{precision + recall} \tag{1}$$

Detections with $f = 0$ are false alarms, whereas detections with $f = 1$ are perfect matches and thus should be most suitable for further description tasks. F-scores of partial overlaps will range between 0 and 1. In the recovery of OOV words, we aim at maximizing the number of detections and their f-scores.

4 Setup

4.1 TED Talks

TED[2] features a collection of more than 600 talks in English language about specialized topics given to a broad audience (`http://www.ted.com`). The vocabulary of many talks is highly specific (e.g. about biology, astronomy, politics . . .) and hence provide a potential source for topic-related OOV words. Manual reference transcripts are prepared in-house and being published as subtitles. Their intent, however, was to preserve correct meaning rather than to provide a word-by-word transcript as often used in the performance evaluation of speech recognition systems. We manually selected 45 talks (10 hours) of various lengths between 5 and 25 minutes.

4.2 Recognizer Setup

Our meeting recognizer developed for NIST Rich Transcription 2007 (RT07) evaluations within the AMI/AMIDA project [7] served as a baseline for reporting word

[2] With permission, TED.com.

Table 1. Word Error Rates using the Baseline Meeting Speech Recognizer and OOV rates using the word/sub-word Recognizer

Talks	Transcript (TED)	Transcript (manual)	OOV rate
All	29.5% WER	-	2.9%
Talk 1	60% WER	41% WER	4.5%
Talk 2	13% WER	13% WER	2.2%

accuracies. It used a 50k language model tuned on lectures, fast speaker adaptations (VTLN, CMLLR) and one-pass bi-gram lattice decoding. Decoding is done on PLP and posterior features processed using HLDA and CVN+CMN. Finally, the bi-gram lattices were expanded to 4-grams. The acoustic models were trained (SAT) on ca. 200 hours of meeting data recorded using independent headset microphones (IHM).

The hybrid recognizer used for partial OOV word detection was derived from this baseline system. The setup was reduced to the first pass (bi-gram decoding only) and the word recognition network was replaced by a hybrid word/sub-word recognition network.

4.3 Hybrid Language Model

The multigram sub-word language model [8] consisted of 3,977 phone and multiphone units trained on the AMI RT06 50k dictionary. The word language model was an open-set Katz-backoff language model trained on a total of 60M words from meetings (AMI/AMIDA, conversational telephone speech (CTS) and broadcast news data (BBC). The interpolation weights for the eight subsets were tuned using seven TED talks. The vocabulary size was fixed to 36k words by frequency cut-off 2 on meeting and CTS data. Due to the large amount of BBC data, this yielded in an uni-gram probability of approximately 1.5% for OOV words and a sensible number of OOV bi-grams. The hybrid language model consisting of a word and a sub-word model was combined in form of weighted finite state transducers by the use of the OpenFST toolkit[3].

4.4 OOV Transcripts

After mapping the reference transcript to unified UK spellings, we ran a forced-alignment to obtain a precise timing of all OOV words. The average OOV rate and word error rate (WER) on all TED talks is shown in the first row of Table 1. To get an impression, how much the available transcripts differ from an ideal word-by-word transcript, we manually transcribed two talks (see last two rows of Table 1). While the WER of talk 1 was notably reduced when using the manual transcript, we found, that the timing of the OOV words did not differ considerably from the TED transcripts.

The minimum and maximum OOV rate on TED talks was 0.3% and 5.1%. Against our assumption, we could not find strong correlation between WER and OOV rate. When checking the forced-aligned transcripts, we found untranscribed speech in some talks, and a few speakers making heavy use of repetitions and hesitations, which were not transcribed at all. This deteriorated word accuracy in many talks, but hardly affected the correct timing of OOV words.

[3] http://www.openfst.org

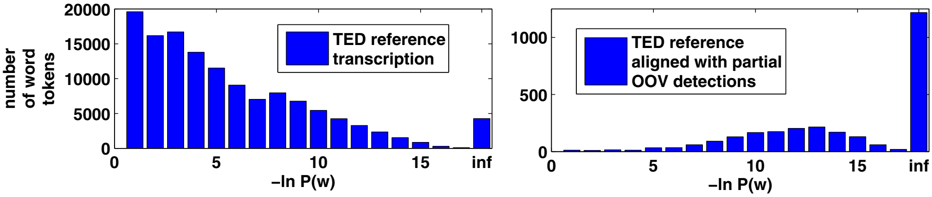


Fig. 2. Word frequency distribution of TED transcripts vs. all partial detections. Word tokens are binned by the negative log likelihood estimate of the language model.

Table 2. Correctness of all recognized words in dependence of their frequency

Frequency Bin	1	2–3	4–7	8–15	16–31	32–63	64–127	128–255	256–511	$\geq$ 512
Correctness (%)	58.8	66.3	68.8	70.2	70.5	71.0	74.4	68.8	67.8	63.2

5 Analysis of Recognition Results

We investigated how the use of the hybrid word/sub-word model excels the recognition of words compared to a word-only recognizer. Therefore, we examine all reference words in TED depending on their estimated and real frequency. Figure 2 compares the word frequency distribution between all words in the TED data and those words, that caused partial OOV detections. It can be seen, that the majority of reference words overlapped with sub-word sequences were actually expected rarely, which is why the word language model estimated rather low likelihoods.

Next, we measured a per-word correctness[4] across all talks. Table 2 shows correctness per exponentially spaced frequency bin. It can be seen, that rarely and frequently recognized words were less often correct than words in the middle frequency ranges. To conclude, hybrid recognition improves recognition for words in the lower frequency regions, where the per-word correctness of the word recognizer is lower.

6 Results of Partial OOV Detection

Furthermore, we examined the quality of partial OOV detection in all TED talks containing 3,789 OOV words. Running on the fixed operating point provided by the one-best recognition output of the hybrid recognizer, where sub-word sequences were interpreted as partial detections, we obtained 2,898 partial OOV detections. By doing so, the system reached a precision of 40.9% and a recall of 31.4% (1,188 hits and 1,710 false alarms, see also table 3).

Figure 3 shows the precision, recall and f-score of all partial OOV detection tokens sorted independently by score. A high number of false alarms are shown in the right with scores of 0. But to the left, a fair number of partial detection tokens overlap almost perfectly.

[4] A recognized word has been considered correct iff there was an overlap in time with the reference word covering all phonemes of the reference word, and both reference and recognized words showed identical spelling.

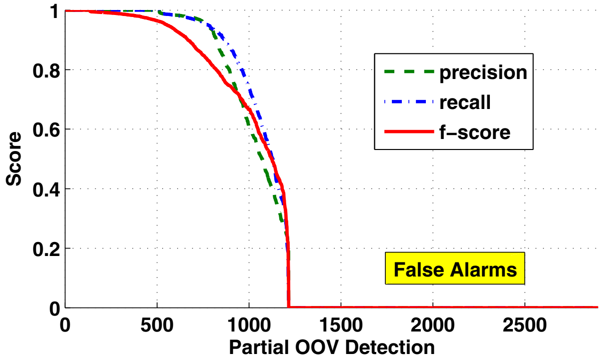


Fig. 3. Score distributions over partial OOV detections sorted by descending score

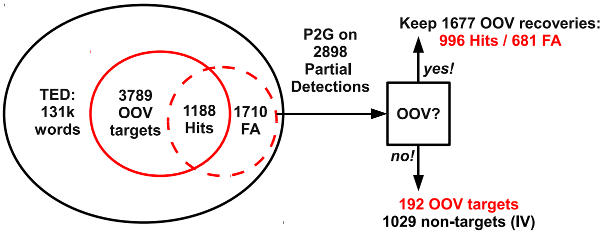


Fig. 4. Improving the Precision of partial OOV detection using P2G Conversion

7 Recovery of Rare Words

Partial OOV detections with high f-scores qualify for word recovery. Using Phoneme-to-Grapheme (P2G) conversion, we successfully recovered the correct spelling of detections from the phonetic description inherent in the corresponding sub-word sequences.

7.1 Setup

We trained a joint multigram P2G model up to 8-grams on the 36k decoding dictionary using Sequitur[5]. Co-alignments of length one and zero between phonemes and characters have been used. We kept 10% of all words for evaluation, where we obtained 21% WER and 5% CER for the generated pronunciations.

7.2 Results

During OOV recovery, we ran P2G on all partial detections, and divided those into two sets according to whether the obtained spelling was an OOV[6] or IV. Figure 4 shows,

[5] http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html

[6] I.e. a spelling which was not contained in our dictionary used for recognition.

Table 3. OOV Detection and Recovery Correction Statistics of Rare Words

3,789 targets/127k non-targets	Total	Hits	FAs	Precision	Recall	Corrected
all OOV detections	2,898	1,188	1,710	40.9%	31.4%	760 (26.2%)
recovered to OOV	1,677	996	681	59.4%	26.3%	277 (9.6%)
recovered to IV	1,221	192	1,029			483 (16.6%)

how recovery of partial OOV detections helped to filter out false alarms by omitting all IV recoveries. The following list shows the most frequent OOV recoveries:

12 ×	MYSOLIUM (MYCELIUM), O'RE (mainly false alarms)
11 ×	GRAVITATIONAL
8 ×	REPLICATOR
6 ×	SELEUM (MYCELIUM), PANDEMIC, GE (mainly false alarms), EXTINCTION, COURTICAL (CORTICAL)

The most frequent recoveries which converted to IV words could be categorized as follows:

- rare words: ORGAN, SEMEN, PSYCHO, PARA, PIPELINES, ...
- pre/suffixes: RE, IN, PRE, PRO, CON, ...
- fillers in between misrecognized words: N, M, S, SE, PH, SH, HA, ...

Table 3 shows the subset statistics in detail: The number of partial detections recovering to OOV was 1677, out of which 996 tokens were hits and 681 false alarms. This corresponded to an increase in OOV detection precision from 40.9% to 59.4% for the sake of recall, which in return decreased from 31.4% to 26.3%.

Furthermore it is shown, how recovery of rare words helped to correct the speech recognition output: approximately one quarter of the remaining hits (277 tokens of 211 types) were recovered to the correct spelling. This corresponds to about 7.3% of the total number of OOV words. In addition, 483 rare IV words of 365 types were recovered to the correct spelling. Altogether, this yielded in a successful recovery of 26.2% of all partial OOV detections, and 0.58% absolute reduction (2% relative) of the overall word error. Furthermore, many detections were found to recover into a readable form which would reveal the true meaning to a human reader. A demonstration of OOV word recovery using ten hours of Fisher telephone calls is available at http://www.lectures.cz/_ted.

8 Conclusion

In our experiments, we showed how hybrid word/sub-word recognition in combination with phoneme-to-grapheme conversion is able to recover rare words. The recovery task also motivated the introduction of partial detections and measuring their quality (f-score). Our experiments resulted in improvements of word accuracy and OOV detection performance. We suggest the hybrid word/sub-word recognition as a mean to improve the ASR accuracy especially on rare, information-rich words. Finally, the recovery of partial, potentially reoccurring OOV words, which get detected with a low f-score only, remains an interesting issue for future research.

References

1. Burget, L., et al.: Combination of Strongly and Weakly Constrained Recognizers for Reliable Detection of OOVs. In: ICASSP (2008)
2. Jiang, H.: Confidence Measures for Speech Recognition: A Survey. Speech Communication 45(4), 455–470 (2005)
3. Bisani, M., Ney, H.: Open Vocabulary Speech Recognition with Flat Hybrid Models. In: Ninth European Conference on Speech Communication and Technology (2005)
4. Yazgan, A., et al.: Hybrid Language Models for out of Vocabulary Word Detection in Large Vocabulary Conversational Speech Recognition. In: ICASSP (2004)
5. Szoke, I., Fapso, M., Burget, L., Černocký, J.: Hybrid Word-Subword Decoding for Spoken Term Detection. In: Proc. SSCS 2008: Speech Search Workshop at SIGIR (2008)
6. Bisani, M., Ney, H.: Joint-Sequence Models for Grapheme-to-Phoneme Conversion. Speech Communication 50(5), 434–451 (2008)
7. Hain, T., et al.: The 2007 AMI(DA) System for Meeting Transcription. Multimodal Technologies for Perception of Humans, 414–428 (2008)
8. Deligne, et al.: Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams. In: ICASSP, Detroit, MI, pp. 169–172 (1995)
9. Hazen, T. J., Bazzi, I.: A Comparison and Combination of Methods for OOV Word Detection and Word Confidence Scoring. In: IEEE Intl. Conference on Acoustics, Speech and Signal Processing (2001)

Enhancing Emotion Recognition from Speech through Feature Selection

Theodoros Kostoulas, Todor Ganchev, Alexandros Lazaridis, and Nikos Fakotakis

Wire Communications Laboratory, Department of Electrical and Computer Engineering,
University of Patras, 26500 Rion-Patras, Greece
tkost@wcl.ee.upatras.gr, tganchev@ieee.org, {alaza,fakotaki}@upatras.gr

Abstract. In the present work we aim at performance optimization of a speaker-independent emotion recognition system through speech feature selection process. Specifically, relying on the speech feature set defined in the Interspeech 2009 Emotion Challenge, we studied the relative importance of the individual speech parameters, and based on their ranking, a subset of speech parameters that offered advantageous performance was selected. The affect-emotion recognizer utilized here relies on a GMM-UBM-based classifier. In all experiments, we followed the experimental setup defined by the Interspeech 2009 Emotion Challenge, utilizing the FAU Aibo Emotion Corpus of spontaneous, emotionally coloured speech. The experimental results indicate that the correct choice of the speech parameters can lead to better performance than the baseline one.

Keywords: affect recognition, emotion recognition, feature selection, real-world data.

1 Introduction

The progress of technology and the increasing use of spoken dialogue systems raise the need for more effective and user-friendly human-machine interaction [1]. Awareness of the emotional state of the user can contribute towards more successful interaction experiences [2].

One of the greatest challenges in the task of emotion recognition from speech is dealing with real-life data and addressing speaker-independent emotion recognition. Real-world speech data differ much from acted speech, once characterized by spontaneous speech and genuine formulations [3]. Thus, results reported for acted speech corpora (accuracy of up to 100%) can not be transferred in realistic conditions with reported performance $< 80\%$ (two-class classification problem) and $< 60\%$ (four-class classification problem) [4].

To this end, various approaches for emotion recognition have been reported. In [5], Callejas & Lopez-Cozar studied the impact of contextual information for the annotation of emotions. They carried out experiments on a corpus extracted from the interaction of humans with a spoken dialogue system. Their results show that both humans and machines are affected by the contextual information. In [6], Iliou & Anagnostopoulos statistically selected a feature set towards studying speaker-dependent and speaker-independent emotion recognition on acted speech corpus. In [7], Seppi et al. reported

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 338–344, 2010.

classification results with the use of acoustic and linguistic features, utilizing the FAU Aibo Emotion Corpus [8,9]. In [10], Ververidis & Kotropoulos optimized the execution time and accuracy of the sequential floating forward selection (SFFS) method in speech emotion recognition. In [11], Brendel et al., describe research efforts towards emotion detection for monitoring an Artificial Agent by voice.

Despite the great effort in the area of emotion recognition the majority of the work conducted offers no chance of compatibility, once no universally-accepted experimental setup had been widely used so far. Though, recent research efforts tend to the establishment [12,13] and utilization of a universally accepted setup [14,15]. The present work reports on-going research activity on affect-emotion recognition within the experimental setup defined by the Interspeech 2009 Emotion Challenge [12]. Specifically we examine the performance of an emotion recognition system in relation with the speech parameters selected for representing the emotional information over five emotion classes. Results indicate that the performance of the system exceeds the baseline performance [12] and is close to the highest performance achieved by [14]. The remaining of this work is organized as follows: Section 2 details the architecture of the emotion recognition system. Section 3 describes the emotional speech data utilized. Experimentations performed, and the results to which these lead are included in Section 4.

2 System Architecture

The block diagram of the GMM-UBM based emotion recognition system is shown in Fig. 1. The upper part of the figure summarizes the training of the speaker-independent emotion models, and the bottom part outlines the operational mode of the system. During both the training and the operational phases, speech data are subject to speech parameterization, which results in a 384-dimensional feature vector [12,16]. The following 16 low-level speech descriptors are computed: zero-crossing-rate (ZCR) from the time signal, root mean square (RMS) frame energy, pitch frequency (normalised to 500 Hz), harmonics-to-noise ratio (HNR) by autocorrelation function and twelve Mel-frequency cepstral coefficients (MFCC) (excluding the 0-th) computed as in the standard HTK setup. For the resulting feature set the delta coefficients are computed. Next, the twelve functionals: mean, standard deviation, kurtosis, skewness, minimum and maximum value, relative position, range as well as two linear regression coefficients with their mean square error (MSE) are applied on sentence level.

As the figure presents, during the training phase two types of data (labelled and unlabeled) are utilized for the creation of the emotion models of interest. Specifically, unlabelled speech recordings, different from the speakers involved in the testing of the emotional models, are utilized for the creation of a large Gaussian mixture model, referred to as Universal Background Model (UBM). This model is sufficiently general not to interfere with any of the emotion categories of interest, and not to represent accurately the individual characteristics of the speakers whose speech was used in its creation. Thus, the UBM is considered to represent emotion-independent distribution of the feature vectors [17,18]. Next, a category-specific set of labelled speech recordings are used for deriving the models for each emotion category of interest. This is done

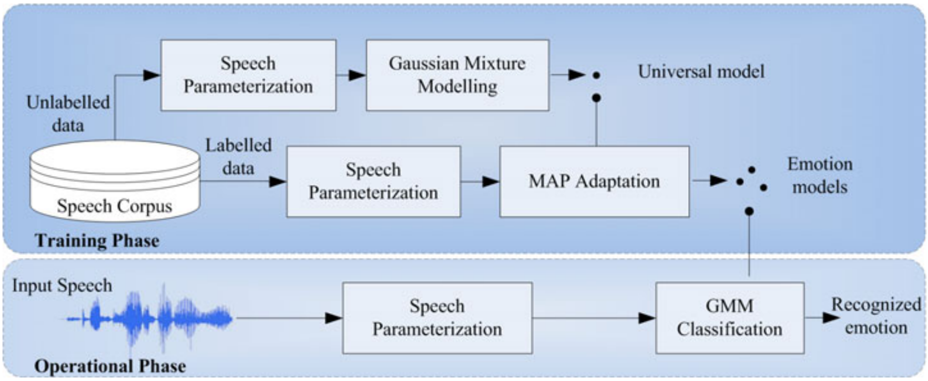


Fig. 1. Block diagram of the emotion recognition component

by the Bayesian adaptation technique also known as maximum a posteriori (MAP) adaption of the UBM [19]. During MAP adaptation only the means were adapted.

The emotion models built during training are utilized in the operational phase for the classification of unlabeled speech recordings to one of the predefined emotional categories. In brief, the feature vectors resulting from the speech parameterization stage are fed to the GMM classification stage, where the log-likelihoods of the input data belonging to each of the category-specific models are computed. Next, these log-likelihoods are subject to the Bayes optimal decision rule, which selects the emotion category with the highest probability.

3 Emotional Speech Data

The present study utilizes the FAU Aibo Emotion Corpus [8] Chap. 5, [9]. The speech corpus results from the interaction of fifty-one 10–12 years old children with Sony's pet robot Aibo, thus consists of spontaneous, emotionally coloured, German speech. The data were collected at two different schools, Ohm (26 children) and Mont (25 children). The robot was controlled by a human operator, causing Aibo to perform a fixed, predetermined sequence of actions; sometimes provoking emotional reactions. The recordings were segmented automatically into turns and annotated on word level by five labellers [8] Chap. 5.

Manually defined chunks based on syntactic-prosodic criteria [8] Chap. 5.3.5 were defined within the Interspeech 2009 Emotion Challenge. The whole corpus consisting of 18,216 chunks. The present work focuses on the five-class classification problem, which considers the following classes: Anger (*angry-touchy-reprimanding*), Emphatic, Neutral, Positive (*motherese, joyful*), Rest. The training dataset consists of the data collected in the Ohm school and the test dataset of the data recorded in the Mont school. In the training set the chunks are given in sequential order with the chunk name indicating the speaker identity. In the test set, the chunks are presented in random order without explicit information about the speaker.

4 Experiments and Results

We split the training dataset specified in Section 3, in two parts: development set and validation set. In each of these parts we preserved the speaker's age and gender distributions similar to those of the entire training dataset. The development set was utilized for performing feature ranking and feature selection experiments. The validation set was utilized for identifying of the most favourable subset of speech parameters and identifying the optimal complexity of the GMM emotion models. The test dataset, as it is defined in Section 3, was used for measuring the accuracy of the emotion recognition system, in the setup defined by the Interspeech 2009 Emotion Challenge.

The selection of the features is performed considering the predictive value of each feature individually, along with the degree of redundancy among them, and for that purpose we relied on the BestFirst search method [20]. 10-fold cross validation protocol was followed. Within this work we selected any speech feature that was selected five or more times out of the 10 evaluations (the 10 splits of data corresponding to the 10 folds). Table 1 shows the 56 speech parameters that were selected, sorted in descending order according to their rank. As can be seen in the table, speech parameters, which are derivatives of the RMS frame energy and MFCCs dominate in the subset top-56 parameters, selected here. These results are in agreement with previous research conducted in the field of emotion recognition [21].

In order to examine the improvement of emotion recognition accuracy contributed by adding more parameters to the speech vector, the following procedure was used: Emotion models were subsequently trained with increasing size of the feature vector, starting from using only the first speech feature and then increasing by one, each time. In order to identify the most appropriate settings of the GMM model, for each size of the feature vector we experimented with different number of mixture components, i.e. $\{1, 2, 4, 8, 16\}$. The optimal accuracy for every feature set, in terms of the un-weighted average (UA) recall computed on the validation set, was observed for a GMM with one mixture component. The UA recall obtained for each of the 56 feature subsets that were evaluated are summarized in Fig. 2. As the figure presents, the maximum UA recall was achieved for the feature vector composed of the first 49 speech parameters.

With the selection of the top-49 speech parameters for the feature vector, and with setting the size of the GMM model to one mixture, the optimization of the emotion recognition system was completed.

On the next step we evaluated the accuracy of the emotion recognition system on the original split of training and test datasets, as formulated in the Interspeech 2009 Emotion Challenge. In detail, utilizing the training set we evaluate the system's performance over the test set using the best performing feature set (top-49) and one mixture for the GMM emotion models. This resulted to 41.99% weighted average recall (accuracy) and 39.45% UA recall. The confusion matrix for this experiment is shown in Table 2.

As Table 2 presents, the broad category Rest (8.6%) was the most difficult to recognize. This can be explained by the large diversity of the data belonging to this category, i.e. this class contains all the data that could not be assigned to one of the other four classes. Category Positive is mostly confused with Neutral, since both subcategories of Positive, i.e. *motherese* and *joyful*, are closer to Neutral than any other category. This can also be observed by the accuracy for Neutral, which is mostly

Table 1. Selected speech parameters

No.	Feature	No.	Feature
1	rmsEnergy-max	29	mfcc1-De-stddev
2	rmsEnergy-min	30	mfcc2-De-max
3	rmsEnergy-range	31	mfcc2-De-min
4	rmsEnergy-mean	32	mfcc2-De-skewness
5	rmsEnergy-stddev	33	mfcc2-De-kurtosis
6	rmsEnergy-kurtosis	34	mfcc4-De-kurtosis
7	rmsEnergy-linregc1	35	mfcc5-De-skewness
8	rmsEnergy-De-max	36	zcr-mean
9	rmsEnergy-De-min	37	zcr-De-stddev
10	rmsEnergy-De-range	38	HNR-max
11	rmsEnergy-De-stddev	39	mfcc3-mean
12	rmsEnergy-De-skewness	40	mfcc11-mean
13	rmsEnergy-De-linregc1	41	mfcc12-mean
14	rmsEnergy-De-linregc2	42	mfcc6-De-maxPos
15	rmsEnergy-De-linregerrQ	43	mfcc8-De-stddev
16	F0freq-De-max	44	mfcc1-kurtosis
17	HNR-De-kurtosis	45	mfcc12-skewness
18	mfcc1-max	46	mfcc4-De-max
19	mfcc1-min	47	F0freq-De-skewness
20	mfcc1-mean	48	mfcc8-kurtosis
21	mfcc1-linregc1	49	mfcc7-De-maxPos
22	mfcc1-linregerrQ	50	rmsEnergy-linregerrQ
23	mfcc3-kurtosis	51	mfcc3-maxPos
24	mfcc4-min	52	mfcc5-max
25	mfcc4-range	53	mfcc9-mean
26	mfcc4-stddev	54	mfcc4-De-linregc1
27	mfcc7-kurtosis	55	F0freq-skewness
28	mfcc1-De-max	56	F0freq-linregerrQ

Table 2. Accuracy in percentages of the optimized emotion recognition system in the setup defined in the Interspeech 2009 Emotion Challenge

	Anger	Emphatic	Neutral	Positive	Rest
Anger	39.1	22.8	19.6	13.1	5.4
Emphatic	14.8	44.3	31.0	5.6	4.3
Neutral	12.9	17.0	44.3	22.1	3.7
Positive	3.3	2.3	27.0	60.9	6.5
Rest	13.4	9.9	36.6	31.5	8.6

confused with category Positive. Moreover, relatively high accuracy of Neutral affective state can be explained by the large number of instances which allows successful adaptation of the universal model towards building the neutral model.

In general, the emotion recognition component shows significant capability on recognizing emotions in all classes but the Rest one, related to the number of

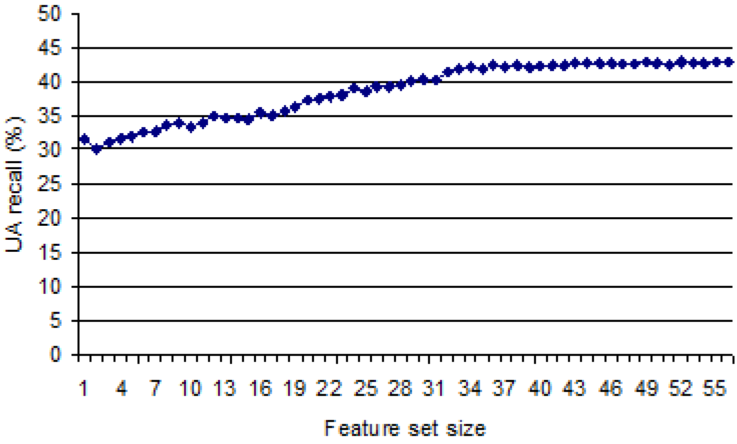


Fig. 2. The UA recall computed on the validation set for feature vectors composed of 1,2,..., 56 speech parameters

available instances in the training set Anger (881), Emphatic (2,093), Neutral (5,590), Positive (674), Rest (721). The emotion recognition system achieves 10% and 3.3% relative improvement to dynamic and static modelling baseline ones provided in [12] respectively and performs close to the highest one reported along the challenge (41.3% UA recall) [14].

5 Conclusion

The present work reported research efforts towards speaker-independent speech emotion recognition, utilizing the setup defined by the Interspeech 2009 Emotion Challenge. Specifically, we addressed the five-class emotion problem, outperforming the baseline accuracy and being close to the highest one reported along the challenge.

The UBM-GMM based emotion recognition system shows significant capability to overcome the class imbalance problem, reporting high value of un-weighted average recall. Correct choice of speech parameters leads to improvement of the system's performance demonstrating the importance of feature selection on the demanding task of recognizing spontaneous, emotionally coloured speech and genuine formulations.

Acknowledgments. This work was supported by the PlayMancer project (FP7-ICT-215839-2007), which is funded by the Seventh Framework Programme of the European Commission.

References

1. Pantic, M., Rothkrantz, L.: Toward an Affect-Sensitive Multi-Modal Human-Computer Interaction. Proc. of the IEEE 91, 1370–1390 (2003)
2. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.G.: Emotion Recognition in Human-Computer Interaction. IEEE Signal Processing Magazine 18(1), 32–80 (2001)

3. Batliner, A., Fisher, K., Huber, R., Spilker, J., Nöth, E.: How to Find Trouble in Communication. Speech Communication 40, 117–143 (2003)
4. Batliner, A., Burkhardt, F., van Ballegooy, M., Nöth, E.: A Taxonomy of Applications that Utilize Emotional Awareness. In: Erjavec, T., Gros, J. (eds.) Language Technologies, IS-LTC 2006, pp. 246–250 (2006)
5. Callejas, Z., Lopez-Cozar, R.: Influence of Contextual Information in Emotion Annotation for Spoken Dialogue Systems. Speech Communication, 416–433 (2008)
6. Iliou, T., Anagnostopoulos, C.N.: Comparison of Different Classifiers for Emotion Recognition. In: 13th Panhellenic Conference on Informatics, pp. 102–106 (2009)
7. Seppi, D., Batliner, A., Schuller B., Steidl, S., Vogt, T., Wagner, J., Devillers, L., Vidrascu, L., Amir, N., Aharonson, V.: Patterns, Prototypes, Performance: Classifying Emotional User States. In: Interspeech 2008, pp. 601–604 (2008)
8. Steidl, S.: Automatic Classification of Emotion-Related User States in Spontaneous Children's Speech. Logos Verlag, Berlin (2009)
9. Batliner, A., Steidl, S., Hacker, C., Nöth, E.: Private Emotions vs. Social Interaction – a Data-driven Approach towards Analysing Emotion in Speech. In: User Modeling and User-Adpated Interaction (UMUAI) 18(1-2), 175–206 (2008)
10. Ververidis, D., Kotropoulos, C.: Fast and Accurate Feature Subset Selection Applied into Speech Emotion Recognition. Elsevier Signal Processing 88(12), 2956–2970 (2008)
11. Brendel, M., Zaccarelli R., Devillers, L.: Building a System for Emotions Detection from Speech to Control an Affective Avatar. In: Proceedings of LREC 2010, pp. 2205–2210 (2010)
12. Schuller, B., Steidl, S., Batliner, A.: The Interspeech 2009 Emotion Challenge. In: Interspeech 2009, ISCA, Brighton, UK, pp. 312–315 (2009)
13. Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Mueller, C., Narayanan, S.: The Interspeech 2010 Paralinguistic Challenge. In: Interspeech 2010, ISCA, Makuhari, Japan (2010)
14. Kockmann, M., Burget, L., Cernocky J.: Brno University of Technology System for Interspeech 2009 Emotion Challenge. In: Interspeech 2009, ISCA, Brighton, UK, pp. 348–351 (2009)
15. Steidl, S., Schuller, B., Seppi, D., Batliner, A.: The Hinterland of Emotions: Facing the Open-Microphone Challenge. In: Proc. 4th International HUMAINE Association Conference on Affective Computing and Intelligent Interaction 2009 (ACII 2009), vol. 1, pp. 690–697 (2009)
16. Eyben, F., Wollmer, M., Schuller, B.: openEAR - Introducing the Munich Open-Source Emotion and Affect Recognition Toolkit. In: Proc. of the 4th International HUMAINE Association Conference on Affective Computing and Intelligent Interaction 2009 (ACII 2009). IEEE, Amsterdam (2009)
17. Reynolds, D.A., Rose, R.C.: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Transactions on Speech and Audio Processing 3, 72–83 (1995)
18. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. Roy. Stat. Soc. 39, 1–38 (1977)
19. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing 10, 19–41 (2000)
20. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
21. Schuller, B., Batliner, A., Seppi, D., Steidl, S., Vogt, T., Wagner, J., Devillers, L., Vidrascu, L., Amir, N., Kessous, L., Aharonson. V.: The Relevance of Feature Type for the Automatic Classification of Emotional User States: Low Level Descriptors and Functionals. In: Interspeech 2007, ISCA, Antwerp, Belgium, August 2007, pp. 2253–2256 (2007)

Collection and Analysis of Data for Evaluation of Concatenation Cost Functions*

Milan Legát and Jindřich Matoušek

University of West Bohemia in Pilsen, Faculty of Applied Sciences,
Department of Cybernetics, Univerzitní 8, 306 14, Plzeň, Czech Republic
{legatm,jmatouse}@kky.zcu.cz

Abstract. This paper describes the collection and analysis of data, which are planned to be used for the evaluation and development of concatenation cost functions for unit selection based TTS systems. Data, collected via listening tests following the recommendations given in [1], were analyzed in a variety of ways to identify and possibly exclude "malicious" listeners as well as to demonstrate their sufficient "richness" for the aimed utilization. This study was limited to five Czech vowels as these sounds are characterized by being highly energetic and having rich spectral content, which induces complexity and wide range of possible discontinuities at concatenation points.

Keywords: TTS, unit selection, concatenation cost, listening tests.

1 Introduction

Unit selection based concatenative speech synthesis still represents an approach that, without question, produces synthetic speech of the highest naturalness. The idea of this method is to have more than one instance of each unit stored in a large speech database and to search at runtime for the best sequence of units to generate the desired utterance.

In order to select the best sequence of units, two cost functions are typically calculated – *target cost* and *concatenation (join) cost*. While the task of the target cost function is to estimate the perceptual difference between a target and a candidate unit, the concatenation cost function should reflect a level of perceived discontinuity between two consecutive units.

The concatenation cost consists mostly of a set of sub-components associated with a difference in pitch, energy and spectra of adjacent segments of concatenated units. A weak point of the concatenation cost functions is the spectral component as no objective measure seems to correlate well with human perception of discontinuities in spectra.

A large number of methods have been proposed in last decade, but none of them proved to be comparatively better than others across all languages and recording conditions. The presented results have sometimes even been in contradiction. Thus, the design of the concatenation cost functions is still an open issue, and there is a lot of work remaining to be done.

* Support for this work was provided by the Grant Agency of the Czech Republic, project No. GACR 102/09/0989, by the Ministry of Education of the Czech Republic, project No. 2C06020, and partly also by the University of West Bohemia in Pilsen, project No. SGC-2010-054.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 345–352, 2010.

Generally, there are two ways of evaluating the concatenation cost functions. One can have candidates for a concatenation cost function, synthesize a set of sentences using each of them separately, and then ask listeners to choose the best version. This method is however quite laborious and costly. The other and mostly preferred option is to simply concatenate some units, let a group of listeners assess the quality of the concatenation points and then calculate correlations between values obtained by an objective measure and the listeners' scores.

The crucial point of the latter approach is to have appropriate test stimuli for the listening tests and to collect reliable results based on the listeners' answers. The purpose of this paper is to present an analysis of data we have collected following the recommendations given in [1].

2 Data Collection

2.1 Stimuli

For our experiments we have used recordings covering five Czech short vowels in all consonantal contexts made by two professional speakers (male and female) in an anechoic room. These speakers had been previously found to be appropriate candidates as they had recorded the corpora for the unit selection based TTS system [2] as well as corpora intended for the first experiments with a limited domain Czech emotional speech synthesis [3].

Recorded data were re-synthesized using the "half sentence" method described in [1] resulting in a huge set of sentences, containing only one concatenation point in the middle each. Our preliminary analyses revealed that a difference in pitch and energy at concatenation points is an important factor to be taken into consideration in order to obtain data exploitable for the design and evaluation of the concatenation cost functions, strictly speaking, the spectral component of these functions.

Note that in our preliminary listening test, all perceptually discontinuous concatenation points were clearly separable from the continuous ones in the $F0$ difference × Energy difference plane, resulting in a database which would not be feasible for an evaluation of the spectral component of concatenation cost functions. In most related studies, the standard procedure is to smooth the concatenation points with respect to differences in pitch and energy to ensure that any perceived discontinuity is not due to pitch or energy "jumps" at the concatenation points. In our study, we have decided to do concatenations without any post-processing to limit a risk of causing audible signal degradations. As the set of synthesized sentences was quite huge, we have rather measured these differences and taken them into account when selecting candidates for our listening test stimuli.

The selection of sentences included into the listening test stimuli was based on methods summarized in Tab. 1. Subsets `f0B`, `enB` and `efB` were included to confirm that large differences in pitch and energy at concatenation points are a significant source of perceived discontinuities. When selecting the candidates for the `f0B` set, a limit for the difference in energy was set to 1dB. For the selection of candidates for the `enB` set, only sentences where the difference in pitch was less than 10 mels were used. The subset `efB` consists of sentences with the largest Euclidean distance from the origin in the $F0$ difference × Energy difference plane. The pitch and energy differences were calculated pitch synchronously having all recordings previously pitch marked using the Multi-Phase Pitch-Mark Detection Algorithm [4]. Since large measured differences in pitch and energy at concatenation points very often appear due to phonetic segmentation and/or pitch marking errors, all candidate sentences have been checked manually, and erroneous sentences were excluded.

Based on the results presented in [1], we put more stress on sentences with smooth pitch and energy transitions at concatenation points, i.e. the subsets `efS`, `mfS`, `beS`, `mfB`, `beB`. The subset `efS` consists of sentences with the smallest Euclidean distance from the origin in the $F0$ difference × Energy difference plane. For the selection of candidates for the other four subsets, we ranked all sentences according to the Euclidean distances from the origin in the $F0$ difference × Energy difference plane and took into consideration only one third of the best ones. The MFCC based distance was calculated as the Euclidean distance between two standard 39-dimensional MFCC vectors characterizing the left and right one pitch period long segments of the boundary region, respectively. The calculation of the LSM based distance was done in line with the method presented in [5] — the dimension of the SVD was set to 10 and the length of the extraction window was set as K=3.

The total number of sentences presented to listeners in each listening test was 1310. Note that the sentences themselves were not the same for both voices as the selection depended on the actual values measured on the synthesized candidates. However, the words containing the concatenation points (three words per vowel) and the number of sentences in each subset were the same (see Tab. 1).

Table 1. Subsets of sentences contained in the listening test stimuli

Set	Description	Num.
`f0B`	large pitch discontinuity and continuous energy transition	150
`enB`	large energy discontinuity and continuous pitch transition	150
`efB`	large pitch discontinuity and large energy discontinuity	150
`efS`	continuous energy and pitch transition	75
`mfS`	small pitch and energy difference + small MFCC based distance	75
`beS`	small pitch and energy difference + small LSM based distance	75
`mfB`	small pitch and energy difference + large MFCC based distance	225
`beB`	small pitch and energy difference + large LSM based distance	225
`ran`	random selection	135
`nat`	original recordings	15
`rev`	revision sentences	15
`dbl`	same source and target left and right consonantal contexts	20

2.2 Subjects

The subjects were university students, all native speakers of Czech. Some listeners stated that they had some background in phonetics. There were 29 subjects who finished the first listening test (male voice) and 27 subjects in the second one (female voice). Approximately half of the subjects were the same across the 2 tests. All subjects were paid upon completion of the listening tests.

2.3 Procedure

The task of the listeners was to assess the concatenations in the middle vowel of the central word of each sentence on both a five-point scale (*no join at all* – 1, *unnatural but not disturbing* – 2, *slightly perceived join* – 3, *highly perceived join* – 4, and *highly disturbing join* – 5) and a binary scale (*perceived join* or *not perceived join*). To make the task easier, natural versions of the middle words containing the concatenation points were played to the listeners prior to synthesized sentences.

Both listening tests were conducted using a web interface allowing the listeners to work from home. It was, however, stressed in the test instructions that the tests shall be done in silent environment and using headphones. It is, of course, clear that organizing the tests in a laboratory would provide us with more consistent testing conditions, but taking into account the number of listeners and the length of the whole listening test, it would be unacceptably time-costly. To gain more control over the listeners, we have not only analyzed logs from our test server but also included some control mechanisms into the tests themselves (see the analysis below).

Based on the lessons learned presented in [6] and also on the feedback we had collected in our preliminary listening test, we have provided the listeners with some examples of discontinuities to help them with calibration for the more fine grained scale. They were instructed to undergo the calibration phase before starting the listening test itself and after each break they made. The logs from the test server confirmed that they indeed did so. It was allowed to listen to the calibration sentences at any time during the listening test. There were no restrictions on how many times a listener played a sentence before assessing it.

3 Analysis of Listeners' Answers

Generally speaking, listening tests are still the most reliable way of assessing the quality of synthetic speech. Nevertheless, the key factor which is not completely under control are the listeners themselves. In order to estimate consistency of their assessments and to identify possibly non-reliable participants, the detailed analysis described in the following sections has been undertaken.

3.1 Checking the Listeners' Consistency and Reliability

As found in [1] and also in [6], the inclusion of natural speech samples is a valuable resource for identifying "malicious" listeners. The test stimuli for both of our listening tests contained 15 – three words per vowel – completely natural sentences (`nat` subset). Using these sentences, we have estimated ability of the listeners to identify natural speech. Those, who assessed a natural sentence as containing an audible join, were given one penalty point for each such decision.

In addition, the ratings based on the five-point scale were also checked and each listener was given a score using a penalization scheme shown in Tab. 2. We have ranked all listeners according to their performance on this set and identified a small group of deviating listeners.

Table 2. Penalization scheme based on the listeners' answers using the five-point scale. "Diff" stands for a difference in listener's scores given to a `rev` sentence or a difference from zero in the case of `nat` sentences.

Diff	**Penalty**
0	0
1	−0.001
2	−0.01
3	−0.1
4	−1

Another useful measure was based on the `rev` subset, which provided us with an indicator of a listener's consistency. The method of scoring the listeners' assessments given to the sentences contained in this subset was based on a method similar to the scoring based on the `nat` subset. Here, the scores given to two instances of a same sentence were compared, meaning that an inconsistent decision on the binary scale was penalized one point, and any difference in the assessments on the five-point scale was penalized using the penalization scheme shown in Tab. 2. Upon rating the listeners we proceeded in the same way as for the `nat` subset to identify possibly non-serious participants.

It is worth noting at this point that we found some sentences which were quite ambiguous in their nature as they contained hardly perceivable joins. This was actually very likely the explanation for why the performance of the listeners in terms of binary scale assessments was generally worse on the `rev` sentences than on the `nat` subset.

3.2 Distribution of Assessments

When we more closely inspected the listeners' binary scale assessments, we realized that there was a trend in both listening tests – for the male voice approximately 60% of the sentences were assessed as containing an audible join, for the female voice the number was even higher reaching 75%. By contrast, there was a small group of listeners in both tests whose scores were equally distributed or even inversed. We have decided to penalize these listeners as it could suggest that they were not serious enough when completing the listening tests or even providing random answers.

All listeners were instructed to use the full range of categories when assessing the joins on the five-point scale. Thus, it was also interesting to analyze the distribution of their five-point scale answers. The results of the analysis are given in Fig. 1 showing the box plots of ratings distributions. Again, there was a small group of listeners in both tests whose answers were outlying.

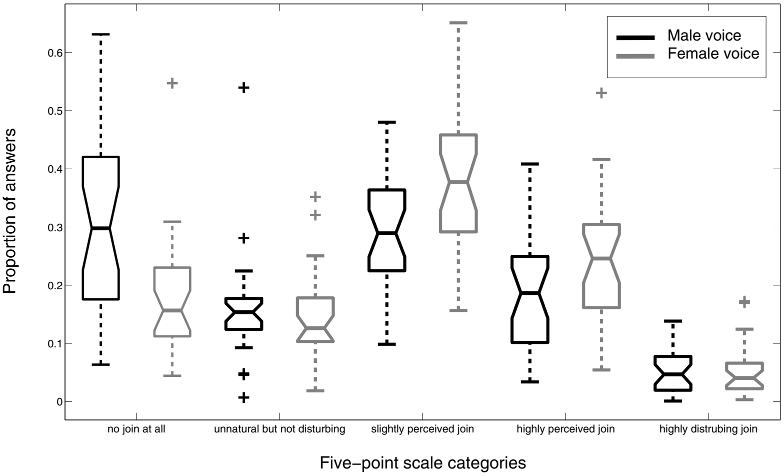


Fig. 1. Distribution of listeners' assessments on the five-point scale

3.3 (Dis)agreement with Facts

The next step of the analysis was to collect "facts", i.e. sentences which were assessed by a majority of listeners in the same way on the binary scale, either as containing an audible join or being completely natural. In our study we have set an ad hoc majority threshold to 80%. After the collection of the "facts", we have started iterating and calculating a disagreement score for each listener using the following formula:

$$\mathrm{DISAGR}_{ij} = \frac{\mathrm{NUM\_DIFF}_{ij}}{\mathrm{FACT\_COUNT}_{j}} \times 100\,[\%]\,, \tag{1}$$

where DISAGR_{ij} is the disagreement score of the i-th listener in the j-th iteration, $\mathrm{NUM\_DIFF}_{ij}$ is a number of assessments of the i-th listener different from "facts" in the j-th iteration and $\mathrm{FACT\_COUNT}_{j}$ is the number of "facts" collected in the j-th iteration.

Starting with all listeners, we have performed a few iterations to see the course of a metric based on the DISAGR measure for the worst listener in each iteration (see Fig. 2). In each iteration the worst listener was excluded and a new set of "facts" was collected. It is obvious that this metric allowed us to isolate a group of suspicious listeners in both listening tests (dashed ellipses).

3.4 Correlation with MOS

While the disagreement score (1) was defined to evaluate the listeners' answers based on the binary scale, we turn next to the analysis of the five-point scale ratings. In order to see how consistent the listeners were using this scale, we have performed an analysis of correlations of particular listeners with the group Mean Opinion Score (MOS). We

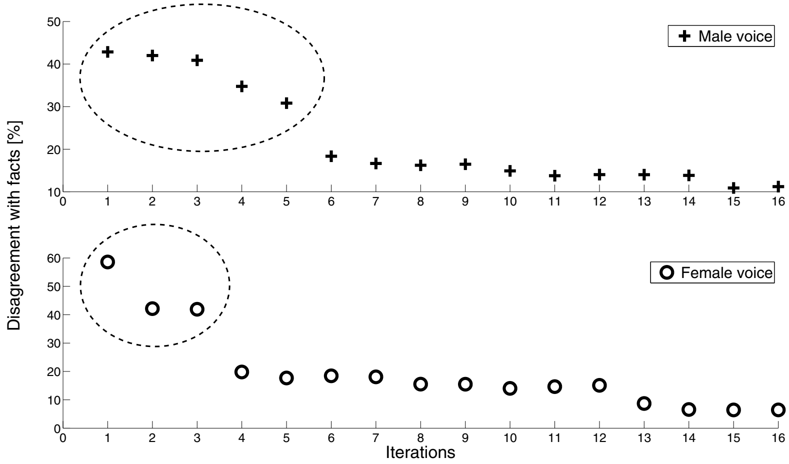


Fig. 2. Metric based on listeners' disagreements with created "facts" (see eq. (1)). The circles and crosses represent the DISAGR value of the worst listener in each iteration.

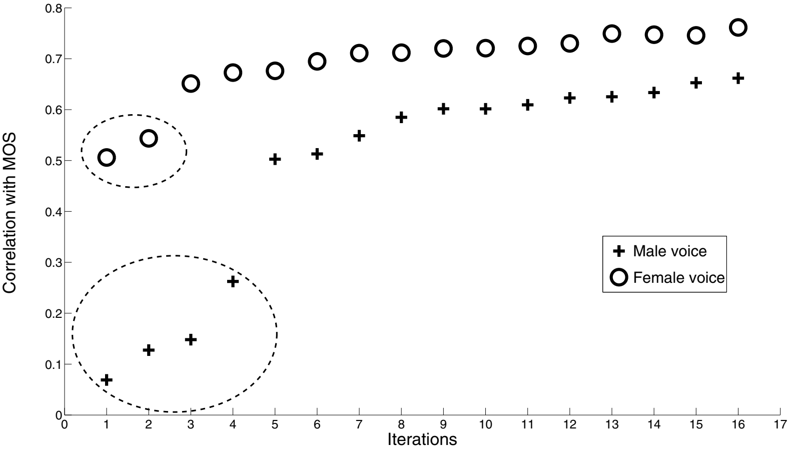


Fig. 3. Metric based on listeners' correlations with the group MOS. The circles and crosses represent the correlation with MOS of the worst listener in each iteration.

have again performed couple of iterations identifying and removing the worst listener from the set, and recalculating the MOS value in each (see Fig. 3).

4 Results

Upon scoring the listeners based on the performed analysis, we have ranked them according to the obtained scores and decided to remove 9 and 6 participants of the male and female voice listening test, respectively. All excluded listeners obtained two or more penalty points, meaning that they were identified as deviating in at least two analysis steps. In the resulting sets the correlation with group MOS of the worst listener was 0.51 for the male voice and 0.67 for the female voice, the disagreement scores of the worst listeners were 15.99% (male voice) and 17.81% (female voice), and the total numbers of collected "facts" were 494 (male voice) and 887 (female voice). For the female voice we have unfortunately collected small number of continuous "facts" (only 11.16%), but these were still mixed with discontinuous "facts" in the small $F0$ difference × small energy difference area suggesting that there was another source of perceived discontinuity (likely the spectrum), which was our goal.

5 Conclusions and Future Work

Based on the analysis steps we have proposed in this paper, we have been able to identify and exclude "malicious" listeners who participated in the listening tests we have conducted to collect data for the design and evaluation of concatenation cost functions for unit selection based TTS systems. Despite collecting only a small number of continuous ratings for the female voice — 11.16% in contrast to 32.79% for the male voice — we still believe that we have collected a feasible dataset as the continuous and discontinuous "facts" were not separable on the pitch and energy difference basis in contrast to study [1].

The future work will focus on the development and evaluation of distance measures for costing of spectral discontinuities at concatenation points in five Czech vowels using the created database. It was also interesting to analyze the distributions of "facts" with respect to the different methods (Tab. 1) used for selecting the candidates for the listening test stimuli, and we plan to report on those results in another paper too.

References

1. Legát, M., Matoušek, J.: Design of the Test Stimuli for the Evaluation of Concatenation Cost Functions. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 339–346. Springer, Heidelberg (2009)
2. Matoušek, J., et al.: Recent Improvements on ARTIC: Czech Text-to-Speech System. In: Proceedings of the 8th International Conference on Spoken Language Processing Interspeech 2004 – ICSLP, Jeju, Korea, vol. 3, pp. 1933–1936 (2004)
3. Grůber, M., Legát, M., Ircing, P., Romportl, J., Psutka, J.: Czech Senior COMPANION: Wizard of Oz Data Collection and Expressive Speech Corpus Recording. In: Human Language Technologies as a Challenge for Computer Science and Linguistics, Wydawnictvo Poznanskie Sp. z o.o., Poznan, pp. 266–269 (2009)
4. Legát, M., Matoušek, J., Tihelka, D.: A Robust Multi-Phase Pitch-Mark Detection Algorithm. In: Interspeech 2007, Antwerp, vol. 1, pp. 1641–1644 (2007)
5. Bellegarda, J.R.: A Novel Discontinuity Metric for Unit Selection Text-to-Speech Synthesis. In: SSW5 2004, pp. 133–138 (2004)
6. Bennett, C.L.: Large Scale Evaluation of Corpus-Based Synthesizers: Results and Lessons from the Blizzard Challenge 2005. In: Interspeech 2005, pp. 105–108. Carnegie Mellon University, Pittsburgh (2005)

Emotion Recognition from Speech by Combining Databases and Fusion of Classifiers

Iulia Lefter[1,2], Leon J.M. Rothkrantz[1,2],
Pascal Wiggers[1], and David A. van Leeuwen[3]

[1] Delft University of Technology, The Netherlands
I.Lefter@tudelft.nl, L.J.M.Rothkrantz@tudelft.nl, P.Wiggers@tudelft.nl
[2] The Netherlands Defense Academy
[3] TNO Human Factors, The Netherlands
david.vanleeuwen@tno.nl

Abstract. We explore possibilities for enhancing the generality, portability and robustness of emotion recognition systems by combining data-bases and by fusion of classifiers. In a first experiment, we investigate the performance of an emotion detection system tested on a certain database given that it is trained on speech from either the same database, a different database or a mix of both. We observe that generally there is a drop in performance when the test database does not match the training material, but there are a few exceptions. Furthermore, the performance drops when a mixed corpus of acted databases is used for training and testing is carried out on real-life recordings. In a second experiment we investigate the effect of training multiple emotion detectors, and fusing these into a single detection system. We observe a drop in the Equal Error Rate (EER) from 19.0 % on average for 4 individual detectors to 4.2 % when fused using FoCal [1].

1 Introduction

Emotion recognition from speech is a field that gains more and more attention from researchers. Typically, machine learning techniques are used to train models of features extracted from databases of emotional speech [2]. Even though the general architectures of the systems are similar, no unity can be found within the components. The results are hard to compare due to inconsistencies in data, task and labeling. Details of these problems are outlined in [3] where a call for standardization is being made.

Obtaining data for training is not trivial. A recent trend is to replace acted emotions by real, spontaneous ones. For this purpose, different emotion elicitation methods are used, e.g. children interacting with a remotely controlled pet robot [4].

Recent work aims at finding the most important feature types [5]. The idea is to extract a large number of features and then reduce this figure, keeping most relevant ones. However, the resulting feature set is highly dependent on the database being used. As noted by [6], different features are relevant in the case of acted and spontaneous emotions.

The goal of this paper is to explore the portability of emotion recognition systems and improve the robustness. Usually experiments involve the use of single databases. As a first experiment, we use four databases of emotional speech, three with acted emotions

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 353–360, 2010.

and one with real-life recordings from call centers. Our approach is to choose a fixed database for testing and to use different data combinations for training. This includes training on the same database, on a different database and on a merged database that includes or not the test database, in a speaker independent way. Typically models are database dependent and are not expected to work well on new types of data. A way to remedy this is to provide a larger amount of training data. With this experiment we examine the benefits of using extended corpora as well as the portability of systems trained on acted data to real life scenarios. Research using multi-corpus training and testing is presented in [7] and [8].

Since the performance of emotion recognition systems is still far from 100% accurate, especially when test data is from a different dataset than the training one, we investigate the improvements of fusing the results of more classifiers trained with different feature sets on spontaneous data. We use both utterance and frame level features, whose combination is expected to enhance the recognition as shown in [9]. Late fusion by linear combination of the scores given equal weights and also weights calculated with logistic regression are compared. Both of them yield higher performance than the individual classifiers.

This paper is organized as follows. In Section 2 we introduce the emotional speech databases used in this work and the methodology for training and testing. Details about the setups and the results of the first and the second experiment are provided in Sections 3 and 4 respectively. The last section contains our conclusions.

2 Methods and Materials

We use four databases for training and testing: the German database (BERLIN) [10], the Danish database of emotional speech (DES) [11], the audio part of the eNTERFACE'05 database (ENT) [12] and the South-African Database (SA) [13]. Details about the characteristics of these databases can be found in Table 1.

The idea is to use subsets of the databases that contain the same emotions. Firstly, we use only combinations of the three acted databases and the three emotions they have in common: anger, happiness and sadness (Experiment 1.a). Secondly, we include also the database of spontaneous speech, and consider just two classes: anger and neutral (Experiment 1.b). Even though ENT does not contain a neutral class, we have decided to use its samples from the anger class in this experiment.

Given a fixed test set, three training conditions are implemented for the first experiment: *within corpus* (same database is used for training and testing), *cross corpus* (the databases for training and testing are different), and *mixed corpus* (samples corresponding to the same emotion but belonging to different databases are considered as one class, and speaker independent classification is performed).

Our approach is to consider one emotion as target and the other emotions as non-target. A detector for the target emotion can make two types of errors that can be traded off: misses and false alarms. We asses the performance of our detectors in terms of equal error rates (EER) where false alarm and miss rates are equal.

All experiments are implemented using speaker independent cross-validation with z-normalization of features on the training set in order to achieve $\mu = 0$ and $\sigma = 1$. For

Table 1. Characteristics of the databases of emotional speech used

Feature	BERLIN	DES	ENT	SA
# anger	127	50	211	1,000
# disgust	38		211	
# fear	55		211	
# happiness	64	52	208	
# sadness	53	52	211	
# surprise		50	211	
# boredom	79			
# neutral	78	52		2,000
# speakers	10 (5 male)	4 (2 male)	46	1,228
acted/spontaneous	acted	acted	acted	spontaneous
language	German	Danish	English	English/Afrikaans
utterance type	preset	preset	preset	free
mean duration (sec)	2.76	5.46	2.81	4.3
total duration (min)	22.8	30.68	59.06	215.02
recording condition	mic	mic	mic	telephone

BERLIN and DES which have a small number of speakers we use leave-one-speaker-out cross-validation. For ENT and SA we use 10 fold cross-validation.

In the case of the second experiment we fuse detectors whose scores span different ranges (some are log likelihoods, some probabilities). It is therefore important to normalize the scores. For this reason we have used 10-fold speaker independent double-cross validation and an adapted form of t-normalization [14]. The mean and standard deviation of the scores of the non-target development set are used in order to normalize the scores of the evaluation set.

3 Experiment 1—Multiple Corpus Training and Testing

In this experiment we test the ability of models trained on one database to generalize to another one. We use a prosodic, utterance level feature set inspired from the minimum required set of features proposed by [15] and the approach of [16]. The feature set contains: pitch(mean, standard deviation, range, absolute slope (without octave jumps), jitter), intensity (mean, standard deviation, range, absolute slope, shimmer), means of the first 4 formants, long term averaged spectrum (slope, Hammarberg index, high energy) and center of gravity and skewness of the spectrum. These features were extracted using Praat [17] and we will refer to them as prosodic features. Classification is performed by Support Vector Machines (SVM) with a radial basis function (RBF) kernel by means of LIBSVM [18]. We refer to this method as SVM.

The results for Experiment 1.a, which uses the three acted databases and three emotions are presented in Table 2. The results for Experiment 1.b in which all four databases and two classes are used are provided in Table 3. The within corpus results can be considered as reference values. In general the cross corpus tests result in worse EERs than the reference values. Interestingly, there are also some exceptions which

Table 2. Results of Experiment 1.a

Experiment	Train corpus	Test corpus	EER		
			anger	happiness	sadness
within corpus	BERLIN	BERLIN	11.6	18.9	14.8
	DES	DES	31.8	33.0	25.0
	ENT	ENT	26.1	36.7	22.3
cross corpus	DES	BERLIN	31.5	53.2	44.3
	ENT	BERLIN	44.9	45.4	19.9
	DES+ENT	BERLIN	38.4	46.8	24.0
	BERLIN	DES	31.9	44.7	33.0
	ENT	DES	**29.9**	34.0	**13.1**
	BERLIN+ENT	DES	**29.9***	34.5	**17.5**
	BERLIN	ENT	38.8	45.6	30.2
	DES	ENT	33.2	36.9	**16.8**
	BERLIN+DES	ENT	35.7	**36.2***	**16.4***
mixed corpus	BERLIN+DES+ENT	BERLIN	20.5	25.0	**3.5**
	BERLIN+DES+ENT	DES	**26.5**	**32.5**	**15.5**
	BERLIN+DES+ENT	ENT	30.1	**36.2**	**16.3**

are printed in bold. Results marked with a star (*) in the cross corpus experiment highlight that there is an improvement by merging databases for training. The mixed corpus approach gives an improvement to both the within- and cross corpus results for most conditions.

Table 3. Results of Experiment 1.b

Experiment	Train corpus	Test corpus	EER
within corpus	BERLIN	BERLIN	1.4
	DES	DES	28.4
	SA	SA	15.5
cross corpus	BERLIN+DES+ENT	SA	29.9
mixed corpus	BERLIN+DES ENT+SA	BERLIN	3.9
	BERLIN+DES+ENT+SA	DES	**25.5**
	BERLIN+DES+ENT+SA	SA	16.5

When all four databases are used, we are interested, for the cross corpus case, in the performance of classifiers trained on acted and tested on real data. In this case the EER of the cross corpus condition is twice that of the within-corpus condition. The mixed approach shows an improvement only in the case of testing on DES, while for SA the result is slightly lower than the reference value. The ENT database is only used in this experiment for training, since it does not contain a neutral class.

4 Experiment 2—Fusion of Classifiers

The aim of this experiment is to improve the performance of emotion detection. We use only the SA database, which is more difficult since it contains free natural speech as opposed to preset utterances and the convenient lab conditions are replaced with noisy telephone speech. We are interested in the performance of different classification methods, as well as their fusion.

A first detection approach uses SVM and the prosodic feature set described in Section 3. Further, we use three spectral feature based classifiers popular in speaker recognition. They are based on Relative Spectral Perceptual Linear Predictive (RASTA PLP) coding of speech [19]. In order to extract the features from the sound signal, voice activity detection is performed based on energy levels. Every 16 ms, 26 coefficients are extracted for a frame of 32 ms : 12 PLP coefficients plus log energy and their derivatives.

The Universal Background Model—Gaussian Mixture Model (UBM-GMM) [20] approach models each class by a mixture of Gaussians based on the RASTA PLP features. We use a 512 mixtures precomputed (UBM) trained on NIST SRE 2008 data. This is MAP adapted using either emotion or neutral speech data. We refer to this method as GMM.

The third technique is a UBM-GMM-SVM detector [21]. The feature supervectors are the means of the UBM-GMM model. These feature sets are used for SVM classification.

The final classifier is known as dot-scoring (DS) [22], and it is a linear approximation of UBM-GMM. It uses sufficient fixed size zero and first order statistics of these features. The method includes channel compensation, meaning that the impact of the communication medium is reduced.

Two types of score level fusion are applied on the scores of these four classification methods: a linear combination of the t-normalized scores with equal weights, and fusion by calculating the weights using linear logistic regression using FoCal [1]. For the second fusion type, a constant is added to the formula for calibration. This approach provides simultaneous fusion and calibration in a way that optimizes discrimination and calibration. The fused scores tend to be well-calibrated detection log-likelihood-ratios.

As we expect different classifiers based on different features to complement each other, we fuse in turn the SVM with prosodic features with each of the GMM-like approaches which are based on RASTA PLP features. Finally, we fuse all four classifiers by logistic regression. The results are shown in Table 4 for different weights of each classifier to the final result. The weights different than 1 are calculated with FoCal. SVM gives the highest performance from the individual classifiers and is always assigned high weights for fusion. However, UGS which has a lower performance by itself is assigned slightly higher weights.

The detection error tradeoff (DET) curves [23] of the individual detectors and their fusion are presented in Figure 1. The results show clearly that fusion leads to great improvements.

Table 4. EERs for individual classifiers and various combinations with different weights

Classifier	Weight										
GMM	1				1	3.03					3.35
UGS		1					1	5.62			5.92
DS			1						1	1.81	1.72
SVM				1	1	5.77	1	5.44	1	5.27	5.13
EER(%)	21.2	19.8	19.6	15.5	11.3	9.9	10.5	10.1	11.3	11.0	4.2

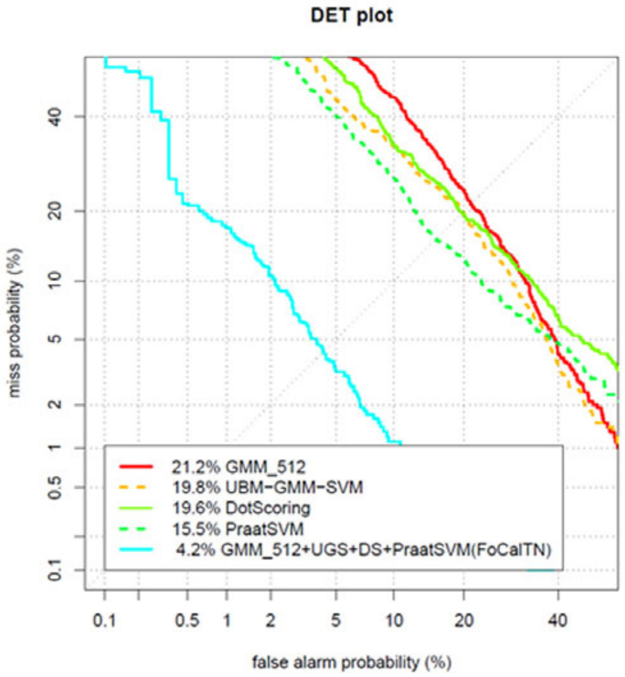


Fig. 1. DET curves for the fusion of Dot Scoring, SVM, GMM and UBM-GMM-SVM by logistic regression

5 Conclusions

In this paper we have investigated several aspects related to emotion recognition in speech. First, we have investigated the ability of an emotion detector to generalize to a different data set. For this we use the common emotions found in three widely used emotion databases: BERLIN, DES, and ENT with emotions anger, happiness and sadness. Surprisingly, we found that for the DES test data we obtained better performance for detectors trained on data including the ENT database, than using training on DES alone. This may be due to the fact that when using DES for training, only 3 actors are available, of which only 1 has the same gender as the test speaker. Here, the classifier obviously can benefit from a wider variability in speakers, even if the recording protocol, way of eliciting the emotions, or even the language of the speech used are different. Another aspect of emotion in speech is whether it results from acted or real emotion. In order to study this, we used data collected from a call center, where two emotions dominate calls from clients: anger and neutral. Here, we observe that mixing in acted emotions does not lead to additional performance with our baseline classifier. We may presume that the emotion cause (real versus acted) is too different between the test and additional training data, although we cannot exclude that other sources of variation (channel, language) also prevent the emotion models from improving.

As a final experiment we have looked into methods for improving a single emotion detector tested on the natural SA data. By using additional features and several frame-based classifiers borrowed from speaker recognition, we could show a very strong improvement in performance from 19 % on average for the 4 individual detectors to 4.2 % for the fused detectors. This is consistent with what has been observed in speaker recognition [1], but it is interesting to note that the fusion still works so well for very short duration utterances (2 seconds, compared to the 2 minutes we are used to in speaker recognition) and for three classifiers that are based on the same spectral features. Although there still is a long way to go before we have robust emotion detectors that are not sensitive to spontaneity, language or recording channel, we do believe that the methods and experiments presented in this paper give some insight into what can be promising approaches in both technology and data collection.

References

1. Brümmer, N., Burget, L., Cernocky, J., Glembek, O., Grezl, F., Karafiat, M., van Leeuwen, D.A., Matejka, P., Schwarz, P., Strasheim, A.: Fusion of Heterogeneous Speaker Recognition Systems in the STBU Submission for the NIST Speaker Recognition Evaluation 2006. IEEE Transactions on Speech, Audio and Language Processing 15(7), 2072–2084 (2007)
2. Pantic, M., Rothkrantz, L.J.M.: Towards an Affect-Sensitive Multimodal Human-Computer Interaction. Proceedings of the IEEE, 1370–1390 (2003)
3. Schuller, B., Steidl, S., Batliner, A.: The INTERSPEECH 2009 Emotion Challenge. In: Proceedings of Interspeech, pp. 312–315. ISCA (2009)
4. Steidl, S.: Automatic Classification of Emotion-Related User States in Spontaneous Children's Speech, 1st edn. Logos Verlag, Berlin (2009)
5. Batliner, A., Steidl, S., Schuller, B., Seppi, D., Vogt, T., Wagner, J., Devillers, L., Vidrascu, L., Aharonson, V., Kessous, L., Amir, N.: Whodunnit – Searching for the Most Important Feature Types Signalling Emotion-Related User States in Speech. Computer Speech and Language (2010)
6. Vogt, T., Andre, E.: Comparing Feature Sets for Acted and Spontaneous Speech in View of Automatic Emotion Recognition. In: IEEE International Conference on Multimedia and Expo., pp. 474–477 (July 2005)
7. Shami, M., Verhelst, W.: Automatic Classification of Expressiveness in Speech: A Multi-corpus Study. Speaker Classification II: Selected Projects, 43–56 (2007)
8. Vidrascu, L., Devillers, L.: Anger Detection Performances Based on Prosodic and Acoustic Cues in Several Corpora. In: LREC 2008 (2008)
9. Vlasenko, B., Schuller, B., Wendemuth, A., Rigoll, G.: Combining Frame and Turn-Level Information for Robust Recognition of Emotions within Speech. In: Proceedings of Interspeech (2007)

10. Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., Weiss, B.: A Database of German Emotional Speech. In: Proceedings of Interspeech, pp. 1517–1520 (2005)
11. Engberg, I. S., Hansen, A. V.: Documentation of the Danish Emotional Speech Database (DES). Internal AAU report, Center for Person Kommunikation (1996)
12. Martin, O., Kotsia, I., Macq, B., Pitas, I.: The eNTERFACE'05 Audio-Visual Emotion Database. In: 22nd International Conference on Data Engineering Workshops (2006)
13. Lefter, I., Rothkrantz, L.J.M., Wiggers, P., van Leeuwen, D.A.: Automatic Stress Detection in Emergency (Telephone) Calls. Int. J. on Intelligent Defence Support Systems (2010) (submitted)
14. Auckenthaler, R., Carey, M., Lloyd-Thomas, H.: Score Normalization for Text-Independent Speaker Verification Systems. Digital Signal Processing 10, 42–54 (2000)
15. Juslin, P.N., Scherer, K.R.: Vocal Expression of Affect. In: Harrigan, J., Rosenthal, R., Scherer, K. (eds.) The New Handbook of Methods in Nonverbal Behavior Research, pp. 65–135. Oxford University Press, Oxford (2005)
16. Truong, K.P., Raaijmakers, S.: Automatic Recognition of Spontaneous Emotions in Speech Using Acoustic and Lexical Features. In: Popescu-Belis, A., Stiefelhagen, R. (eds.) MLMI 2008. LNCS, vol. 5237, pp. 161–172. Springer, Heidelberg (2008)
17. Boersma, P.: Praat, a System for Doing Phonetics by Computer. Glot International 5(9/10), 341–345 (2001)
18. Chang, C. C., Lin, C. J.: LIBSVM: a Library for Support Vector Machines (2001)
19. Hermansky, H., Morgan, N., Bayya, A., Kohn, P.: RASTA-PLP speech analysis technique. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 121–124 (1992)
20. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing 10, 19–41 (2000)
21. Campbell, W., Sturim, D., Reynolds, D.: Support Vector Machines Using GMM Supervectors for Speaker Verification. IEEE Signal Processing Letters 13(5), 308–311 (2006)
22. Brümmer, N.: Discriminative Acoustic Language Recognition via Channel-Compensated GMM Statistics. In: Proceedings of Interspeech. ISCA (2009)
23. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The Det Curve In Assessment Of Detection Task Performance. In: Proceedings Eurospeech 1997, pp. 1895–1898 (1997)

Emologus—A Compositional Model of Emotion Detection Based on the Propositional Content of Spoken Utterances

Marc Le Tallec[1], Jeanne Villaneau[2], Jean-Yves Antoine[1],
Agata Savary[1,4], and Arielle Syssau[3]

[1] LI, Université de Tours, France
{marc.letallec,jean-yves.antoine,agata.savary}@univ-tours.fr
[2] VALORIA, Université de Bretagne Sud, France
jeanne.villaneau@univ-ubs.fr
[3] Université de Montpellier 3, France
arielle.syssau@univ-montp3.fr
[4] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. The ANR EmotiRob project aims at detecting emotions in an original application context: realizing an emotional companion robot for weakened children. This paper presents a system which aims at characterizing emotions by only considering the linguistic content of utterances. It is based on the assumption of compositionality: simple lexical words have an intrinsic emotional value, while verbal and adjectival predicates act as a function on the emotional values of their arguments. The paper describes the semantic component of the system, the algorithm of compositional computation of the emotion value and the lexical emotional norm used by this algorithm. A quantitative and qualitative analysis of the differences between system outputs and expert annotations is given, which shows satisfactory results, with the right detection of emotional valency in 90% of the test utterances.

Keywords: Detection of Emotions, Child SLU, Emotional Norm.

1 Introduction

A new important field of study in robotics is the domain of companion robots which execute complex tasks and offer behavior enrichment through their interaction with human beings. The French project EmotiRob, supported by the ANR (National Agency of Research), belongs to this research domain and aims at conceiving and realizing a “reactive” autonomous soft toy robot, which can emotionally interact with children weakened by disease, and give them some comfort. Previous experiments have shown the contribution of companion robots in this type of situation.

In the EmotiRob project, the robot simulates emotional states by facial expressions. To enable it to simulate a pertinent emotion, our system aims at detecting emotions conveyed in words used by children by combining prosodic and linguistic clues. Only the latter ones are addressed in our present study, i.e., we rely merely on the propositional content of a child’s utterance.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 361–368, 2010.

2 Detection of Emotions

There is currently no consensus about what an emotion is and how an emotion has to be characterized. An emotion is a complex cognitive state, which is strongly dependent on various contexts: short-term context includes the type and the circumstances of the interaction, while long-term context is related to cultural and personal life. We resume the two approaches which are most used to characterize emotions. In the first one, emotions are classified into emotional modalities. The set of modalities may vary but most of authors agree to a classification into seven emotional modalities: anger, disgust, enjoyment, fear, surprise, sadness and neutrallity [1,2]. The second approach uses an ordinal classification in a muldimensional space. For example, some psycholinguists use excitement level and emotional valency (negative/positive). All these works show both of the following conclusions:

1. In a real dialogue, most of the speech turns do not convey perceptible emotion, as 80% of them are classified as neutral from an emotional point of view.
2. The perception of the emotions is very variable. The measures of the inter-annotator agreement give poor results. A referential annotation may be achieved with a majority vote only.

Most of the time, emotion detection performs classifications, by using acoustic or prosodic clues. The use of linguistic clues such as indications of repairs or presence of emotional words is not frequent, although this use improves the performances of the systems [3]. Besides, these performances are still perfectible.

Because linguistic emotion detection seems a hard task and has hardly been investigated, we have chosen to represent emotion as a simple pair (valency, intensity), where valency can be negative, positive or neutral and where intensity is measured by an integer from 0 to 2.

Before realizing an automatic system of emotion detection based on those principles, it was necessary to know if the task had a chance of success, Therefore we first tested how good the agreement of annotators can be on emotions conveyed by the lexical content of sentences produced by children. Choosing a representative test corpus is not easy, since an interactive emotional robot for children does not exist currently. We collected a corpus in a primary school, where about 7-year-old children were asked to invent tales. The corpus (so-called Brassens corpus) is composed of about 170 sentences which make up twenty short stories. Two annotations were performed by nine (5 adults and 4 children) annotators: in the first one, sentences were given in a random order, so that an out-of-context annotation can be obtained, while in the other one the sentences were given in the order of the stories. Agreement between annotators is calculated by using Kappa coefficients, which show low correlation between the child annotators (0.49 for the out-of-context and 0.38 for the contextual annotation), however the various ages (from 4 to 9) of the four annotators can partially explain these poor results. On the other hand, there is a good agreement between adult annotators (0.86 for the out-of-context and 0.84 for the contextual annotation), an encouraging result within the framework of our purpose.

3 Natural Language Understanding

Our objective is to detect emotions which are conveyed in the propositional content of a linguistic message. One can presume—and this assumption has been confirmed by our surveys (see Section 5)—that many words convey a positive or negative emotion by themselves. Therefore, a simple first approach to obtain an emotional measure of a message is to add up the emotional measure of each of its words. An advantage of this solution is that no understanding of the message is required. An evaluation of the emotional potential of each domain word has to be achieved only. Our baseline is based on this principle.

Nevertheless, it is obvious that this solution does not work in all cases. For example, *"la mort de la méchante sorcière" (the dead of the mean witch)* does not convey a negative emotion although each of its words does. In this example, it can be assumed that the emotional potential of the concept *"death"* is dependent on its related object. Based on these principles, our work aims at realizing a compositional calculus of the emotional content of the utterances. To achieve that, a semantic treatment of the utterances is required, which achieves their "understanding" i.e. specifying the semantic linkages between concepts.

Spoken Language Understanding (SLU) is a difficult task, especially because of speech recognition errors and spoken disfluencies. A very robust parsing is required and current operational systems are related to restricted tasks with a restricted vocabulary [4,5,6]. Thus, the objective of trying to understanding the utterances of children within the framework of the EmotiRob project may seem unattainable. To make the task feasible, the domain has been restricted to the concepts of the world of very young children, 4 or 5 years old. Moreover, complete understanding is not always necessary: the baseline may replace predicative calculus in the case of partial understanding.

Our SLU system is based on logical formalisms and performs an incremental deep parsing [7]. It provides a logical formula to represent the meaning of the word list that Automatic Speech Recognition provides to the SLU as input. The understanding module performs a translation from natural language to a target logical formalism. The vocabulary known by the system as the source langage contains about 8,000 lemmas selected from the lexical Manulex[1] and Novlex[2] bases. We have restricted the concepts of the target langage by using Bassano's studies related to the development of child language [8]. SLU carries out a projection from the source langage into Bassano's vocabulary.

The parsing is split into three main steps: the first step is chunking [9] which segments the sentence into minimal syntactic and semantic groups. The second step builds semantic relations between the resulting chunks and the third is a contextual interpretation. The second and third step use a semantic knowledge of the application domain. Thus, the main work that had to be done to adapt the system to our objective was to build an ontology from the set of application concepts, a difficult task due to the width of the application domain. More precisely, Bassano vocabulary includes many verbs, some of them with polysemic meaning. To specify the possible uses of these

[1] `http://leadserv.u-bourgogne.fr/bases/manulex/manulexbase/indexFR.htm`

[2] `http://www2.mshs.univ-poitiers.fr/novlex/`

verbs, a part of the ontology [10] is based on a linguistic corpora study related to fairy tales.

Figure 1 shows the parsing of the utterance: *"Il était une fois un petit cochon qui n'avait pas d'amis" (Once upon a time there was a little pig who had no friends).* Chunking provides six chunks which are gradually linked in the following parsing steps. The logical formula provided by the system is:

(narrative (neg (to_have [(subject: (pig [(size: little)])), (object: (friends))])))

The calculus related to emotion detection of this utterance is given in the following sections.

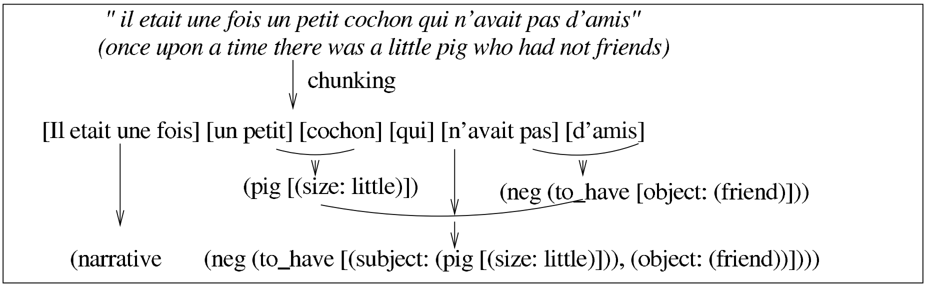

" il etait une fois un petit cochon qui n'avait pas d'amis"
(once upon a time there was a little pig who had not friends)
chunking
[Il etait une fois] [un petit] [cochon] [qui] [n'avait pas] [d'amis]
(pig [(size: little)])
(neg (to_have [object: (friend)]))
(narrative
(neg (to_have [(subject: (pig [(size: little)])), (object: (friend))]))

Fig. 1. An example of parsing

4 Basic Principles of the EMOLOGUS System

In the EMOLOGUS system, the detection of emotions relies on a major principle: the emotion conveyed by an utterance is compositional. It depends on the emotion of every individual word as well as the semantic relations characterized by the SLU system. More precisely, simple lexical words have an intrinsic emotional value, while verbal

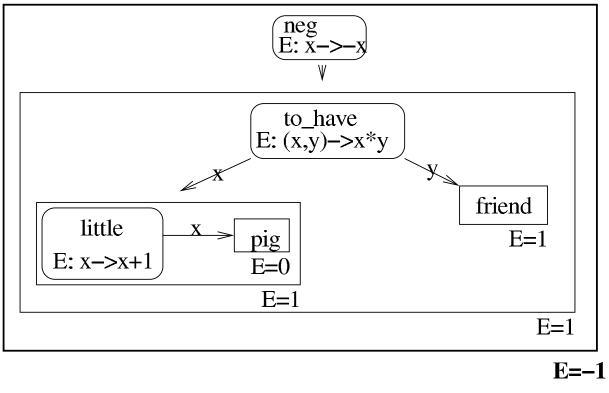


Fig. 2. Compositional calculus for the example sentence (cf. Fig. 1)

and adjectival predicates act as a function on the emotional values of their arguments. As an illustration, consider the sentence of Fig. 1 and its related logical formula. The computation of the emotion begins with the consideration of the emotional value of the words *pig* and *friends* ($E = 0$ for *pig* and $E = 1$ for *friends*), which are simple arguments of the formula. Then, adjective such as *little* and verbs such as *to have* acts as predicate on these initial values. For instance, *little pig* is assigned $E = +1$ as emotional value, since *little* is defined as the predicate *E:* $x \mapsto x + 1$. The successive applications of the predicates provide the global emotional value of the sentence: $E = -1$ (Fig. 2). As an illustration, here are different definitions of predicates found in our lexicon:

$E : x \mapsto x + 1$ (aimable / kind) $E : x \mapsto x - 1$ (énervé / irritated)
$E : (x, y) \mapsto -y$ (casser / to break) $E : (x, y) \mapsto 1$ (chatouiller / to tickle)
$E : (x, y) \mapsto min(x, y)$ (accompagner / to go with)

5 Emotional Norm and Definition of Predicates

The first requirement is to know emotional valency which is associated to the lexicon words by children. This has been the aim of emotional lexical standards which have been used for a long time in experimental psychology. These standards compile subjective evaluations of a population of judges about one or several emotional characteristics of words.

Some standards are related to emotional characteristics such as the duration of the emotion caused by a word [11,12]. However in all emotional lexical standards, two characteristics are always estimated: valency and intensity.

Both characteristics are mainly estimated by an adult population, using nominal scales of judgment (positive, neutral, negative) or ordinal (i.e., -5 very negative through 5 very positive). To our knowledge, only two standards have compiled the evaluations made by young children: Vasa et all [13] for the English language, and Syssau and Monnier [14] for the French language (5, 7 and 9-year-old children). In these standards, the answer scales used are the same as those used with adults with slight modifications. The number of modalities is reduced (3 for the study of Syssau and Monnier) and every answer modality is related to a drawing which represents a smiling, sad or neutral face, respectively. The examination of the results shows that from 5 years of age, the children are able to judge emotional valency of the words with a substantial agreement.

In the EmotiRob project, we complete the standard of Syssau and Monnier by the evaluation of emotional valency of 80 new words extracted from the Bassano lexicon with children between 5 and 7 years old. In the original standard, the words were classified by age of acquisition and most of them were common nouns. For the 5-year-old children, the added words, most of them adjectival or verbal, have been divided into 2 lists of 40 words estimated in two different sessions. For those of 7 year olds, the list was to be estimated in a single session. At every age, two random orders of presentation of the words was defined, every order being presented to half of the participants. These experiments were carried out in 4 French schools.

To complete the characterization of our lexicon, an emotional predicate has been assigned to every verb or adjective of our application lexicon through an agreement procedure among five adult experts. More precisely, every expert proposed one or at most two definitions for every predicate. Then, agreement was sought among these proposals. It is interesting to note that it has finally been possible to reach a complete agreement.

6 Experiments and Results

We conducted several experiments in order to assess the behaviour of our system. These experiments have been carried out on the Brassens corpus (cf. Section 2), by using annotations of 5 experts as test references. This evaluation assessed the detection of emotion without considering the discursive context. This is why the test sentences have been provided in a random order to the annotators, who had to describe the emotion valency conveyed by a single scalar value including valency and intensity between –2 (very negative) and 2 (very positive). The reference was obtained through a majority ballot among the expert annotations. Finally, we compared the semantic EMOLOGUS system with the basic baseline presented in Section 3 on 173 emotionally annotated sentences. The results are shown in the following table:

	Baseline	EMOLOGUS
Precision	68.8%	90%

With a precision of 90%, EMOLOGUS presents a good accuracy, by opposition with the baseline. This result suggests that the detection of emotions should greatly benefit from the consideration of the linguistic content, in addition to a standard prosodic analysis.

Table 1. Matrix of confusion for EMOLOGUS and the baseline, respectively

$$\text{EMOLOG.}\backslash\ \text{Ref.}= \begin{array}{r|ccccc|} & -2 & -1 & 0 & 1 & 2 \\ \hline -2 & 4 & 2 & 0 & 0 & 0 \\ -1 & 2 & 18 & 0 & 0 & 0 \\ 0 & 1 & 5 & 116 & 2 & 0 \\ 1 & 0 & 0 & 3 & 16 & 1 \\ 2 & 0 & 0 & 0 & 1 & 2 \\ \hline \end{array} \qquad \text{Baseline}\backslash\ \text{Ref.}= \begin{array}{r|ccccc|} & -2 & -1 & 0 & 1 & 2 \\ \hline -2 & 4 & 0 & 0 & 1 & 0 \\ -1 & 3 & 12 & 7 & 0 & 0 \\ 0 & 0 & 6 & 90 & 4 & 0 \\ 1 & 0 & 4 & 18 & 11 & 1 \\ 2 & 0 & 0 & 7 & 3 & 2 \\ \hline \end{array}$$

Table 1 presents the error confusion matrices of the two systems, which enable an in-depth analysis of the error distribution. Within the EmotiRob context, the most serious error of the system is giving an opposite valency, because it can infer a bad reaction of the robot. This type of error, called "valency inversion", is never observed with EMOLOGUS. Most of its errors correspond to "emotion deletions" errors, when the system does not detect an emotion which is present in the test reference. The opposite error, "emotion insertion" only concerns 18% of the errors made by EMOLOGUS. The last kind of error is less serious: it corresponds to "intensity errors" when the system detects the correct valency but assigns an erroneous intensity (for instance: "positive" vs. "very positive"). Fortunately, 35% of the errors of EMOLOGUS are intensity errors. If one ignores such moderate errors, the precision of the system rises up to 94%. To the contrary, the baseline leads to more serious errors, among which are valency inversions.

Remarks Related to the Analysis of Some Errors

Some verbs naturally have a positive or negative emotion when we do not know who is doing the action, for example in *"La femme était enfermée dans une prison" (the lady was locked in a jail). transl. "To be locked in a jail"* is very negative when the subject is positive, and very positive when the subject is negative. When it is not possible to know who is locked up, a low negative emotion is felt by annotators. It is the same for instance for *can't do something*, *don't believe someone* and the opposite for *find something*. A solution would be to define a default emotional behaviour when the valency of an argument is unknown or neutral.

Some errors result from an erroneous modelisation of adjective or adverb of degree, as *little*. On the whole, this adjective involves a general tendency to shift the valency of its argument to more positive values, as in *a little wolf.* This is why it has been modelled by the predicate: *E:* $x \mapsto x + 1$. Unfortunately, the influence of the adjective should differ in some specific situations.

However, the human annotators do not always respect this behavior. Consider the sentence *"il a vu une petite maison" (transl. "he has seen a little house").* Since *maison/house* does not support any emotion (E = 0), the nominal group *petite maison/little house* should be considered as slightly positive (E = 0 + 1 = 1). However, the majority of our experts consider it as neutral. This means that in some circumstances that still have to be investigated, the adjective does not affect the emotional valency of its arguments. As a result, *little* should present different emotional behaviors. This is a good example of what we can call an emotional ambiguity. Fortunately, the latter seems to be moderate.

We also have problems with elements which directly depend on the context of the story. For instance: *Les parents s'enfuirent (transl. The parents ran away).* Here, we can't be sure if it is positive or negative. The emotional interpretationof such a sentence depends highly on the discourse context: did the parents succeeded in avoiding a danger, or did they have doing something wrong. Naturally, annotators have annotated it with a neutral emotion, while the semantic model chose a positive emotion due to the positive valency of the noun *parents*. For more precision we need more information about why they run away.

7 Conclusion and Perspectives

We have tested our semantic model on a corpus, and results are encouraging. These experiments show it is possible to detect emotion on the basis of linguistic clues with a high precision (90%). A very positive fact is that we never find opposite valency. We have modified the emotional function of some predicates, because of the particularities we have found in these experiments, in particular in sentences involving emotionally neutral subjects. We have to verify these results on a larger corpus, before working on sentences in context, with management of anaphora. Also, we have to think about how to combine emotions in several sentences and about dynamics of emotion on a complete text.

References

1. Ekman, P.: Patterns of Emotions: New Analysis of Anxiety and Emotion. Plenum Press, New York (1999)
2. Cowie, R., Cornelius, R.: Describing the Emotional States that Are Expressed in Speech. Speech Communication 40, 5–32 (2003)
3. Schuler, B., et al.: The Relevance of Feature Type for the Automatic Classification of Emotional User States. In: Interspeech 2007, Anvers, Belgique, pp. 2253–2256 (2007)
4. Seneff, S., Polifroni, J.: A New Restaurant Guide Conversational System: Issues in Rapid Prototyping for Specialized Domains. In: International Conference on Spoken Language Processing (ICSLP 1996), Philadelphia, pp. 665–668 (1996)
5. Glass, J.: Challenges for Spoken Dialogue Systems. In: Proceedings IEEE ASRU. Workshop, Keystone, Colorado, USA (1999)
6. Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., Hetherington, L.: JUPITER: Telephone-Based Conversational Interface for Weather Information. IEEE Transactions on Speech and Audio Processing XX(Y), 100–112 (2000)
7. Villaneau, J. and Antoine, J-Y.: Deeper Spoken Language Understanding for Man-Machine Dialogue on Broader Application Domains: A Logical Alternative to Concept Spotting. In: Proceedings of SRSL 2009, the 2nd Workshop on Semantic Representation of Spoken Language, Athens, Greece, pp. 50–57 (2009)
8. Bassano, D., et al.: Le DLPF, un nouvel outil pour l'évaluation du dévelopement du language de production en français. Enfance 2(5), 171–208 (2005)
9. Abney, S.: Parsing by Chunks. In: Berwick, R., Abney, S., Tenny, C. (eds.) Principle Based Parsing. Kluwer Academic Publishers, Dordrecht (1991)
10. El Maarouf, I.: Natural Ontologies at Work: Investigating Fairy Tales. In: Corpus Linguistics 2009, Liverpool, G.B. (2009)
11. Niedenthal, P.M., Auxiette, C., Nugier, A., et al.: A Prototype Analysis of the French Category "émotion". Cognition and Emotion 18, 289–312 (2004)
12. Zammuner, V.L.: Concepts of emotion: Emotioness and Dimensional Rating of Italian Emotion Words. Cognition and emotion 12(2), 243–272 (1998)
13. Vasa, R.A., Carlino, A.R., London, K., Min, C.: Valence Ratings of Emotional and Non-Emotional Words in Children. Personality and Individual Differences 41, 1169–1180 (2006)
14. Syssau, A., Monnier, C.: Children's Emotional Norms for Six Hundred French Words. Behavior, Research, and Methods 41, 213–219 (2009)

Automatic Segmentation of Parasitic Sounds in Speech Corpora for TTS Synthesis*

Jindřich Matoušek

University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics
Univerzitní 8, 306 14 Plzeň, Czech Republic
jmatouse@kky.zcu.cz

Abstract. In this paper, automatic segmentation of parasitic speech sounds in speech corpora for text-to-speech (TTS) synthesis is presented. The automatic segmentation is, beside the automatic detection of the presence of such sounds in speech corpora, an important step in the precise localisation of parasitic sounds in speech corpora. The main goal of this study is to find out whether the segmentation of these sounds is accurate enough to enable cutting the sounds out of synthetic speech or explicit modelling of these sounds during synthesis. HMM-based classifier was employed to detect the parasitic sounds and to find the boundaries between these sounds and the surrounding phones simultaneously. The results show that the automatic segmentation of parasitic sounds is comparable to the segmentation of other phones, which indicates that the cutting out or the explicit usage of parasitic sounds should be possible.

Keywords: parasitic speech sound, speech synthesis, unit selection, HMM, automatic phonetic segmentation.

1 Introduction

Contemporary *concatenative speech synthesis* techniques based on a *unit-selection framework* employ very large speech corpora. As the principle of unit-selection-based speech synthesis is to select the largest suitable segment of natural speech in order to prevent the potential discontinuities in the connected speech signal [1], attributes like the voice identity, style of speaking, speaking habits, the quality of speaking, etc. are copied to synthetic speech. In order to produce speech as natural as possible, source utterances in the speech corpora have to be spoken naturally, i.e., among others, with a natural intonation, natural speech rhythm or common pronunciation (and, more over, depending on the resulting application possibly also in various expressive or affective states, emotions, etc.). As a result, due to affectedness, carelessness or possibly also hypercorrectness related to the natural way of speaking, the recorded utterances can include so called *parasitic speech sounds* (parasitic from the point of view of both Czech canonical pronunciation and the fluency and overall acceptability of synthetic speech), linguistically non-systematic phenomena. Such sounds, if used in text-to-speech (TTS) synthesis too often or, even, unintentionally, can negatively affect the acceptability,

* This research has been supported by the Grant Agency of the Czech Republic, project No. GAČR 102/09/0989.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 369–376, 2010.

fluency (or, in other words, naturalness) of synthetic speech and can have an intrusive effect on listeners, especially when neutral, unmarked synthetic speech (which is still required in a majority of TTS applications) is about to be produced.

It is obvious that, due to the enormous size of present speech corpora employed in unit-selection-based speech synthesis (usually more than 10 hours of speech), manual annotations of parasitic sounds are almost impossible. Thus, the parasitic sounds are hidden in the corpora and, following the principle of concatenation-based unit selection speech synthesis, they can unintentionally get into synthesised speech. Even worse, when such parasitic sounds are not detected in the source recordings, speech contexts in which the parasitic sounds could appear are to be synthesised with no a priori information about the presence of such a sound. As a result, the speech contexts both with and without the described phenomena could be concatenated, which will be most likely perceived as a discontinuity in synthetic speech. Having information about the presence/absence of a parasitic sound in a given context, one can avoid mixing such speech contexts in unit selection speech synthesis — if the position of the parasitic sound is known, it could be cut out of the speech signal, or the particular speech unit containing the parasitic sound could be penalised during the unit selection mechanism, or, even, such a unit could be intentionally used in speech synthesis in order to increase the naturalness of synthetic speech in some applications.

In [2], the phonetic analysis and identification of parasitic speech sounds were carried out, and a procedure for the identification and *automatic detection* of the presence of parasitic sounds in speech signals was designed. In this paper the next step in the process of the precise localisation of parasitic sounds is presented. The objective is to propose an algorithm for the *automatic segmentation* of parasitic sounds in speech signals. The paper is organised as follows. Parasitic speech sounds are briefly introduced in Section 2. In Section 3, the results of the automatic detection of the parasitic sounds are shown. Experiments with the automatic segmentation of parasitic sounds, results and their discussion are provided in Sections 4 and 5. Finally, conclusions are drawn in Section 6.

2 Parasitic Speech Sounds in Czech

For the purpose of our study, randomly selected recordings of two source speakers (one female, one male) used in the Czech TTS system ARTIC [3], in total approx. 28 minutes of read speech (see Table 1 for more detailed description) were utilised. The recordings were analysed with the aim to identify parasitic sounds — sounds whose fine phonetic detail cannot be regarded as part of the canonical sounds pattern in Czech and whose presence in synthetic speech may negatively affect the perceptibility of synthetic speech.

The results of the phonetic analyses are summarised in the lower part of Table 1. The presence of *glottalization* (preglottalization and postglottalization) turned out to be the most frequent non-standard phenomena in the analysed sample. Glottalization can be defined as a short aperiodic noise produced by the vocal folds. In Czech, one of the phonetic realisations of glottalization is *glottal stop*, which naturally occurs only before a post-pausal vowel [4]. In this context, glottalization is perceived as a natural part of pronunciation. Thus, glottal stop is a well-established unit in the phonetic

Table 1. Description of the speech material in reference and test data sets (used in the context of the automatic detection and segmentation techniques explained in Sections 3 and 4): number of utterances, amount of data in minutes, length in phones and occurrences of the most frequent parasitic phenomena

	male			female		
	all	ref.	test	all	ref.	test
utterances	119	70	49	88	58	30
amount [min.]	13.75	8.84	4.91	15.04	11.34	3.70
phone length	9,850	6,298	3,552	9,979	8,010	1,969
preglottalization	123	73	50	74	53	21
postglottalization	45	16	29	4	0	4

system of Czech and is used in synthesis of Czech speech [3]. On the other hand, the occurrence of glottal stops in preconsonantal positions (almost exclusively after a pause), or *preglottalization*, is not usual in Czech, and it may be viewed as marked, and potentially intrusive. Similarly, *postglottalization*, aperiodic activity of the vocal folds before a pause (either after a consonant or a vowel), is perceived as non-standard and intrusive. For more details see [2] and [5] where phonetic analysis is described more thoroughly, and other parasitic phenomena like epenthetic schwa are discussed as well. Although, as revealed by the phonetic analysis, these sounds do occur in Czech speech, they are not included in the standard Czech phonetic inventory and they have not been coped with in the synthesis of Czech speech yet.

3 Automatic Detection of Parasitic Sounds

The aim of the automatic detection of parasitic sounds was to detect, or identify the presence of the parasitic sounds (preglottalization and postglottalization in our case) in speech signals. Two different kinds of classifiers were used: an HMM-based classifier and BVM classifier. Both types of classifiers were trained on the reference (training) data set and evaluated on the test data set specified in Table 1.

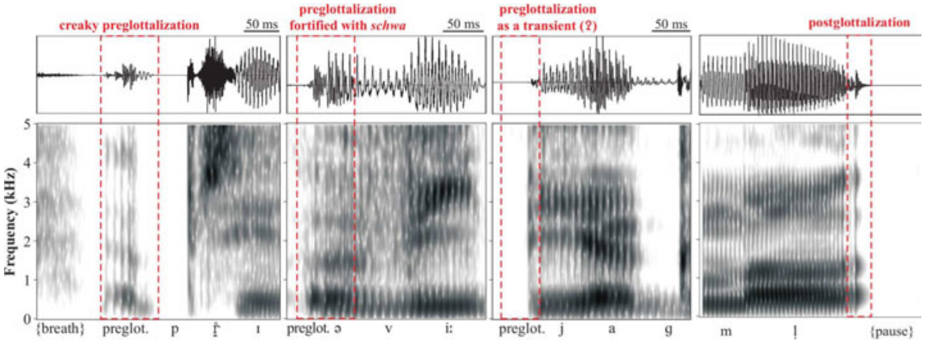


Fig. 1. Examples of parasitic sounds: preglottalization and postglottalization

Table 2. Results of the automatic detection of parasitic sounds [2]

	Preglottalization				Postglottalization			
Detection	male		female		male		female	
measures	HMM	BVM	HMM	BVM	HMM	BVM	HMM	BVM
P	50	50	21	21	26	26	4	4
N	56	59	28	29	106	132	60	64
TPR	0.92	0.92	0.81	0.52	0.77	0.96	0.0	0.75
FPR	0.11	0.02	0.07	0.00	0.02	0.00	0.00	0.00
ACC	0.91	0.95	0.88	0.80	0.94	0.99	0.94	0.98
chance level	0.50	0.51	0.52	0.54	0.70	0.73	0.94	0.90
κ	0.81	0.91	0.75	0.56	0.70	0.98	0.00	0.85

The HMM-based classifier follows the well-established techniques known from the field of automatic speech recognition (ASR) and automatic phonetic segmentation (APS), see e.g. [6,7], or for Czech [8,9]. In this framework each phone or sound is modelled by an hidden Markov model (HMM): firstly the parameters of each HMM are estimated, and then *force-alignment* based on Viterbi decoding is performed to find the best alignment between the HMMs and the corresponding speech data. As this classifier was utilised also for the automatic segmentation of parasitic sounds, it is described further in Section 4 in more detail.

Ball Vector Machines (BVM) classifier, one from the family of kernel methods, was used with RBF (radial basis function) kernel. The TRAPS parametrisation technique with the setup similar to [10] was employed to obtain the input features for the classifier. The parameters of the BVM classifier were determined using grid-search algorithm with 10-fold cross-validation. More details and reasons why this classifier was preferred over the similar ones, CVM or SVM, could be found in [2].

The evaluation of the automatic classification was performed in the "standard" way, i.e. using true positive rate (TPR, i.e. hit rate), false positive rate (FPR, i.e. false alarm rate) and detection accuracy $ACC = [P \cdot TPR + N \cdot (1 - FPR)]/(P + N)$, where P is the number of "positive examples" in the test data (i.e. how many times the parasitic sound really occurred in the given context) and N is the number of "negative examples" in the test data. In order to take also the classification "accuracy" occurred by chance into account, Cohen's kappa κ is also indicated (in our case, $\kappa = 1$ means perfect performance of a classifier, $\kappa \leq 0$ indicates worse performance than that obtained by random classification — generally, $\kappa \geq 0.70$ is considered satisfactory). Results of the detection are summarised in Table 2.

4 Automatic Segmentation of Parasitic Sounds

Though the accuracy of the detection of the HMM-based classifier is slightly worse when compared to the BVM classifier (see Table 2), one of the advantages of the HMM-based classifier is that, as boundaries between HMMs are produced during the alignment, the position of each modelled sound in the utterance could be located. Therefore, the HMM-based classifier was used for the automatic segmentation of parasitic sounds.

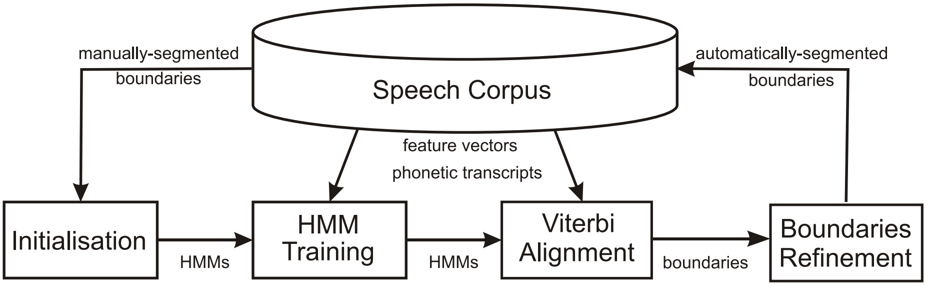


Fig. 2. Simplified scheme of HMM-based automatic phonetic segmentation

In our experiments, a set of single-speaker three-state left-to-right context-independent multiple-mixture HMMs corresponding to all Czech phones and parasitic sounds was employed in a similar way as in the automatic phonetic segmentation task in [9]. For models parameters estimation, we employed isolated-unit training utilising Baum-Welch algorithm with model boundaries fixed to the hand-labelled ones (the reference data). For each utterance from the test data (described by feature vectors of mel frequency cepstral coefficients extracted each 4 ms), the trained HMMs of all phones and parasitic sounds were concatenated according to the phonetic transcripts of the utterance and aligned with a speech signal by means of Viterbi decoding. In this way, the best alignment between HMMs and the corresponding speech data is found, producing a set of boundaries which delimit speech sounds belonging to each HMM. Thus, the position of each phone-like unit and parasitic sound is identified in the stream of speech signal. Within this process, the automatic detection of the presence of each parasitic sound mentioned in Section 3 is carried out by creating multiple phonetic transcripts per utterance with all combinations of the presence/absence of the given parasitic sound in the defined contexts. Consequently, the transcript which "best matches" the data is chosen as the maximum likelihood estimation (MLE) of the utterance. In this way, the parasitic sounds in given contexts could be detected. A simplified scheme of the automatic phonetic segmentation utilising the HMM-based classifier is shown in Figure 2.

The results of the automatic segmentation of preglottalization (PRG) and postglottalization (POG) in terms of mean absolute error (MAE), root mean square error (RMSE) and percentage of boundaries deviating less than the tolerance region 10 ms (Tol10) or 20 ms (Tol20) are shown in Table 3. The results for postglottalization in the female speech corpus are not shown due to the small number of occurrences of postglottalization in female speech signals (see Table 1). Notice that only the ending boundaries of preglottalization (PRG-*) and the starting boundaries of postglottalization (*-POG) are specified. The other types of boundaries (*-PRG and POG-*) are in pauses (see Figure 1), and, due to the smooth concatenation of speech signals in silence, the precise location of these boundaries is not so important. For comparison, the segmentation accuracy of a similar unit, glottal stop (GST-*), and the average segmentation accuracy of all other Czech phonetic units (*-*) are also shown in Table 3. The comparison of the segmentation accuracy of all boundary types for the male speech corpus is shown in Figure 3. Similar results were obtained also for the female corpus.

Table 3. Results of the automatic segmentation of preglottalization and postglottalization sounds in the male and female corpus

	male				female			
Boundary	MAE (ms)	RMSE (ms)	Tol10 (%)	Tol20 (%)	MAE (ms)	RMSE (ms)	Tol10 (%)	Tol20 (%)
PRG-*	7.50	10.56	83.33	90.48	7.13	10.20	66.67	100.00
*-POS	8.33	11.16	65.38	92.31		—		
GST-*	7.00	9.39	73.08	96.15	6.87	12.59	75.00	90.00
-	6.45	11.66	82.40	94.65	6.73	12.58	80.56	94.20

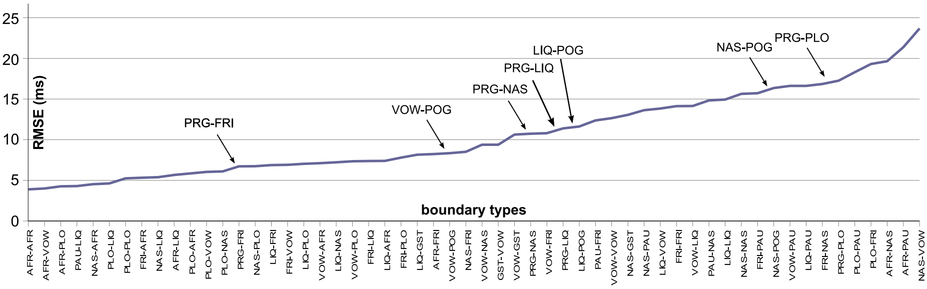


Fig. 3. Comparison of the automatic segmentation accuracy of different boundaries types (VOW = vowels, FRI = fricatives, PLO = plosives, AFR = affricates, NAS = nasals, LIQ = liquids) in terms of RMSE

5 Discussion

Despite the slightly worse results of the automatic detection of the presence of parasitic sounds, the HMM-based classifier was preferred over the BVM classifier in our experiments. It provides us with an "all-in-one" solution — during a single, well-established procedure of the automatic phonetic segmentation, the detection of the presence of parasitic sounds is carried out simultaneously. As a result, the segmentation of all phones and parasitic sounds (if detected in the appropriate contexts) is obtained. Based on these segmentations (boundaries between phones in speech signals), speech unit inventories for unit-selection-based speech synthesis can be built.

Looking at the results of the automatic segmentation in Table 3 and in Figure 3, it can be shown that:

- For both speech corpora, the segmentation accuracy of preglottalization (PRG-*) is comparable to the segmentation accuracy of glottal stop (GST-*), a phonetic unit similar to preglottalization, which has already been used in synthesis of Czech speech (according to the unpaired t–test the difference in MAE is not statistically significant, two-tailed P-value = 0.8095).
- Comparing the segmentation accuracy of (PRG-*) to the average segmentation accuracy of all other phonetic boundaries (*-*), preglottalization tends to be worse

in terms of MAE but it tends to be better in terms of RMSE (with the difference in MAE being not statistically significant, unpaired *t*–test, two-tailed P-value = 0.4876).

- The segmentation of postglottalization (`*-POG`) is less accurate than the segmentation of preglottalization (statistically not significant, unpaired *t*–test, two-tailed P-value = 0.6607).
- Segmentation results in Figure 3 confirm that the segmentation of both preglottalization and postglottalization sounds does not deviate from the segmentation of all other phone sounds.

Moreover, the average segmentation accuracy of the automatic phonetic segmentation (APS) system with the parasitic sounds included (MAE = 6.45 ms, RMSE = 11.66 ms) is better than the average segmentation accuracy of the standard APS system with no parasitic sounds included (MAE = 6.71 ms, RMSE = 16.21 ms), which means that explicit modelling of parasitic sounds does increase the accuracy of the segmentation of other phones (statistically not significant, unpaired *t*–test, two-tailed P-value = 0.5879).

The results indicate that, based on the automatic segmentation, it should be possible to cut the parasitic sounds out of the speech signals and thus to prevent them from getting into synthesised speech. Or, more specifically, parasitic sounds could be used, within reasonable measure, as regular units in speech synthesis in order to increase the naturalness of synthetic speech in some applications.

6 Conclusion

In this paper, the introduction of preglottalization and postglottalization, parasitic speech sounds from the point of view of the fluency and overall acceptability of synthetic speech, was presented. Beside the automatic detection of the presence of the parasitic sounds in speech signals, the research was focused on the automatic segmentation of these sounds in speech. The main goal was to find out whether the segmentation was accurate enough for the parasitic sounds to be cut out of synthetic speech. Alternatively, the precise localisation of their positions in source speech corpora would enable explicit usage of the parasitic sounds in speech synthesis. HMM-based classifier was employed to detect the parasitic sounds and to find the boundaries between these sounds and the surrounding phones simultaneously. The results show that the automatic segmentation of parasitic sounds is comparable to the segmentation of other phones. It indicates that the cutting out or the explicit usage of parasitic sounds should be possible.

In our future work, the utilisation of both proposed classifiers, the BVM one for the detection of parasitic sounds and the HMM-based one for the segmentation of parasitic sounds in the contexts detected by the BVM classifier, will be researched. Furthermore, speech synthesis with the parasitic sounds either excluded or intentionally included will be investigated. The quality of synthetic speech will be compared to the quality of synthetic speech produced by the standard TTS system.

References

1. Tihelka, D., Romportl, J.: Exploring Automatic Similarity Measures for Unit Selection Tuning. In: Proceedings of Interspeech, Brighton, Great Britain, pp. 736–739 (2009)
2. Matoušek, J., Skarnitzl, R., Machač, P., Trmal, J.: Identification and Automatic Detection of Parasitic Speech Sounds. In: Proceedings of Interspeech, Brighton, Great Beritain, pp. 876–879 (2009)
3. Matoušek, J., Tihelka, D., Romportl, J.: Current State of Czech Text-to-Speech System ARTIC. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 439–446. Springer, Heidelberg (2006)
4. Skarnitzl, R.: Acoustic Categories of Nonmodal Phonation in the Context of the Czech Conjunction "a". In: Palková, Z., Veroňková, J. (eds.) AUC Philologica 1/2004, Phonetica Pragensia X, Karolinum, Prague (2008)
5. Machač, P., Skarnitzl, R.: Phonetic Analysis of Parasitic Speech Sounds. In: Proceedings of the 19th Czech-German Workshop on Speech Processing, Prague, Czech Rep., pp. 61–68 (2009)
6. Byrne, W., Doerman, D., Franz, M., Gustman, S., Hajič, J., Oard, D., Picheny, M., Psutka, J., Ramabhadran, B., Soergel, D., Ward, T., Zhu, W.: Automatic Recognition of Spontaneous Speech for Access to Multilingual Oral History Archives. IEEE Transactions on Speech and Audio Processing 4, 420–435 (2004)
7. Toledano, D., Gómez, L., Grande, L.: Automatic Phonetic Segmentation. IEEE Transactions on Speech and Audio Processing 11(6), 617–625 (2003)
8. Vaněk, J., Psutka, J.V., Zelinka, J., Pražák, A., Psutka, J.: Discriminative Training of Gender-Dependent Acoustic Models. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS (LNAI), vol. 5729, pp. 331–338. Springer, Heidelberg (2009)
9. Matoušek, J.: Automatic Pitch-Synchronous Phonetic Segmentation with Context-Independent HMMs. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS (LNAI), vol. 5729, pp. 178–185. Springer, Heidelberg (2009)
10. Schwarz, P., Matějka, P., Černocký, J.: Towards Lower Error Rates In Phoneme Recognition. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 465–472. Springer, Heidelberg (2004)

Adapting Lexical and Language Models for Transcription of Highly Spontaneous Spoken Czech

Jan Nouza and Jan Silovský

Institute of Information Technology and Electronics, Faculty of Mechatronics
Technical University of Liberec, Studentská 2, CZ 461 17 Liberec, Czech Republic
jan.nouza@tul.cz, jan.silovsky@tul.cz

Abstract. The paper deals with the problem of automatic transcription of spontaneous conversations in Czech. That type of speech is informal with many colloquial words. It is difficult to create an appropriate lexicon and language model when linguistic resources representing colloquial Czech are limited to several small corpora collected by the Institute of Czech National Corpus. To overcome this, we introduce transformations between the most frequent colloquial words and their counterparts in formal Czech. This allows us a) to combine the small spoken corpora with much larger corpora of more formal texts, b) to optimize the recognizer's lexicon, and c) to solve the data sparsity problem when computing a probabilistic language model. We have applied this approach in the design of a system for transcription of spontaneous telephone conversations. Its recent version operates with accuracy about 48% and the proposed transformations together with corpora mixing contributed to 9% improvement compared to the baseline system.

Keywords: Speech recognition, colloquial speech, language modeling.

1 Introduction

Automatic transcription of telephone conversations is one of the most challenging tasks in speech recognition research. There are several reasons that make this task very complex. Here are the most relevant ones: a) a telephone signal is bandwith-limited and thus less intelligible, b) many codecs used in digital telephony introduce additional distortion, c) telephone calls usually contain a lot of background noise, d) conversational speech is informal and spontaneous, and e) linguistic context is shared among the speakers, which means, that it is partly missing in individual utterances.

Research in this field has been stimulated by several facts. First, its results are very important, namely for those domains, like national security or police investigations. Second, since 1990s several large corpora containing telephone speech have been collected, e.g. Switchboard [1], CallHome and CallFriend [2] and Fisher corpus [3]. They have been largely used in basic and applied research, in the development of prototype systems and also in evaluation competitions (e.g. [4,5]). Transcription accuracy has improved from initial 40% (in 1990s) to some 60–70%, recently. It should be stated, however, that these figures were reported mainly for English. In other languages they are slightly lower (e.g. about 60% for Dutch in [5]).

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 377–384, 2010.

Automatic transcription of spontaneous Czech is even more complex. Czech is an inflective language with about 10 times larger basic vocabulary compared to English. Moreover, there is also a big difference between formal Czech and its colloquial form used in daily conversations. Furthermore, there exists no corpus of common telephone speech that would be available for researchers. What is available, at least, are several corpora that contain text transcriptions of colloquial Czech. These have been collected by the Institute of Czech National Corpus (CNC) and are available as [6,7].

In this paper, we describe our initial works on developing a system that could provide automatic transcriptions of conversational speech, particularly for telephone calls. We focus namely on its linguistic level, as its acoustic part has been already set up in our previous work [8]. The two main questions to be solved are: Provided there are very limited resources representing the target type of speech, how to construct a lexicon that would be able to cope with colloquial Czech, and how to build a language model for that type of informal and spontaneous conversation?

What we propose is to find links and transformations between the more or less formal written Czech (for which many sources can be found, e.g. on internet) and its oral informal counterpart. By employing these transformations we can increase the amount of data used for building the lexicon and language model applicable for conversational speech.

2 Initial Investigations

We have been working on speech recognition tasks for more than a decade. We have focused mainly on Czech (and in some measure also on several other Slavic languages [9]). The main tasks investigated so far were voice dictation and broadcast speech transcription. In the following text we briefly contrast the results achieved in these two domains to our initial attempts towards telephone speech recognition.

2.1 State of the Art in Czech Speech Recognition

In Table 1, we report typical results achieved by our speech recognition systems in various domains and at different complexity levels (for more details, see [9]).

Table 1. Typical results in speech recognition of Czech in various domains

Task	**Speaking style**	**Lexicon (type and size)**	**Accuracy [%]**
Medical report dictation	dictation	domain-oriented, 130K	97
Legal report dictation	dictation	domain-oriented, 300K	95
Broadcast news (studio)	reading by professionals	general, 350K	94
Broadcast news (street)	reports by professionals	general, 350K	89
Talk shows (studio)	conversational	general, 350K	78

2.2 Initial Experiments in Recognition of Telephone Conversations

For initial investigations we used the existing speech recognition engine[10] with its front-end and acoustic model adapted to 8 kHz telephone signals. Tests were done with 500 telephone utterances (a subset of conversations reported in[8], with the total number of 6,585 words). In the first test, we utilized the general (broadcast-oriented) lexicon and corresponding LM to transcribe them. As expected, the accuracy was poor, just 22.6%. One of the reasons was a rather high Out-of-Vocabulary (OOV) rate (9.4%) due to many colloquial words occurring in reference transcriptions. When we replaced these colloquial words by their In-Vocabulary equivalents, the OOV rate decreased to 1.6% and accuracy increased to 30.7%. The other test was done with a much smaller lexicon and language model created from the two CNC corpora of oral Czech [6,7]. In this case, the accuracy and OOV rates were 25.9% and 7.7%, respectively. These figures (Table 2) show that the latter type of lexicon (though smaller) and the LM were more appropriate as they were derived from the corpora of colloquial Czech. Unfortunately, the size of the CNC corpora is rather small (3 million words). Because of their limited size and scope they do not offer any possibility for further improvements. Therefore, we have to find a way to combine limited oral language resources with those that are available in much larger amount.

Table 2. Initial experiments in telephone conversation recognition with different setup

Lexicon and LM **(size in words)**	**Reference transcription**	**OOV** **[%]**	**Accuracy** **[%]**
General lexicon and LM (350K)	Literal with colloquial words	9.4	22.6
General lexicon and LM (350K)	Colloquial words normalized	1.6	30.7
Lex. and LM - CNC ORAL (80K)	Literal with colloquial words	7.7	25.9

3 Colloquial vs. Formal Czech

There exist several studies that compare formal and colloquial Czech. The recent ones are often based on quantitative observations derived from the previously mentioned CNC corpora, e.g.[11]. Colloquial Czech is the means of communication used in daily conversations by the vast majority of people in Czechia. What makes it different from formal language is, mainly, a) the choice of words, b) modifications in pronunciation, and c) changes in morphology and grammar. In the following text, we focus mainly on phenomenon b) because it is the most frequent one and it can be described by a set of rules. The other two phenomena will be mentioned only briefly.

3.1 Modifications in Pronunciation and Spelling

The oral form of any language always tends to make speaking easier. It is also the case of colloquial Czech. Let us list at least the most frequent types of modifications that have impact also on spelling and hence they introduce new words and word-forms into the language:

Change of "-ý-" into "-ej-". This modification affects almost all adjectives with endings "-ý", "-ým", "-ých", "-ýma". Eg. word "mladý" ("young" in English) has colloquial forms "mladej", "mladejm", "mladejch", "mladejma". The same modification occurs also within some word stems, e.g. "bejt" vs. "být" (eng. "to be") or in prefix "vý-" pronounced and spelled as "vej", e.g. "vejlet" (eng. "trip").

Change of "o-" into "vo-". This phenomenon known as prosthetic "v" adds consonant "v" to almost all words of Czech origin than begin with vowel "o", e.g. "vokno" vs. "okno" (eng. "window"), "omýváš" vs. "vomýváš" (eng. "you wash up").

Vowel shortening in endings "-ám" and "-ím". This affects mainly verbs in 1st person singular. Words, like "myslím" (eng. "I think"), or "dělám" (eng. "I do") get shorter form "myslim", or "dělam", respectively.

Reduced verb endings "-ají" and "-eme". Verbs with these endings in plural 1st and 3rd person are often pronounced and spelled without the last vowel, e.g. "dělaj" (eng. "they do"), or "chcem" (eng. "we want").

Changes in verb endings "-uji" and "-ují". Most verbs with these endings in 1st person singular and 3rd person plural get colloquial form "uju" and "ujou".

Endings "-ama", "-ema", "-ýma". These endings occur in informal declination of nouns and adjectives. Often, they replace formal suffixes "ami", "emi", "ými", but more and more they are used for almost all nouns and adjectives in plural instrumental, regardless of their gender.

Reduction in consonant clusters, like "kd-", "kt-", "js-" or "vžd-". This affects words like "když" → "dyž" (eng. "when"), "který" → "kerý" ("which"), "jsem" → "sem" ("I am"), "jsme" → "sme" ("we are"), or "vždyť" → "dyť" ("but").

The above mentioned examples are just some of the most common modifications that occur in colloquial Czech and that in fact introduce more than twenty thousand "new" words into the lexicon of spoken Czech. Note that most of these words achieve very high frequency in conversational speech and hence they cannot be ignored.

3.2 Introduction of New Words

Many words in colloquial Czech have their origin either in shortening long words, e.g. "spořka" instead of "spořitelna" (eng. "saving bank"), compacting multi-words into single ones, e.g. "hlavák" instead of "hlavní nádraží" (eng. "main station"), expanding abbreviations, e.g. "cédéčko" instead of "CD", or importing words from slang, dialect and foreign languages. This phenomenon can hardly be described by simple linguistic rules. Therefore, we haven no option other than to include the most frequent words of this type in the lexicon.

3.3 Changes in Grammar

Colloquial Czech tries to simplify many grammatical rules that exist in formal language but are too complex to be applied in spontaneous speech. It often happens that there is wrong grammatical agreement between nouns, adjectives and verbs. In this case, usually existing words and word-forms are used but they occur in wrong relations or incorrect positions. Unlike the previous two phenomena, this one has no impact on the lexicon but it has a rather severe influence on the language model.

Table 3. Lexicon structured into 3 levels: normalized form, alternative forms, pronunciations

Normalized form	Alternative forms	Pronunciation forms
milion	milion	milijon
	milión	milijón
mladý	mladý	mladí
	mladej	mladej
omývᚠ	omývᚠ	omíváš
		omíváž
	omejváš	omejváš
		omejváž
	vomejváš	vomejváš
		vomejváž

4 Lexicon Structure Applicable for Colloquial Language

As we demonstrated in Section 2, the main problem in the recognition of colloquial speech is the lack of sufficiently large corpora from which we could extract a representative lexicon and compute a N-gram language model. To solve this problem, we propose a scheme that allows us to combine resources targeted both on formal as well as on informal language.

The proposed lexicon uses three levels. On the highest one, there is the normalized form of a lexicon item. On the second level, there could be several additional versions of the word that differ in spelling, but have the same lexical and semantic meaning. These additional forms reflect either some evolutional changes in spelling (something like "colour" and "color" in UK and US English), or the modifications due to colloquial background. On the third level, there is pronunciation for each of the items. An example of this structured lexicon is shown in Table 3. Notice, that on the pronunciation level, there can be also several alternatives, for example, in cases when phonetic assimilation can affect the voiced/unvoiced characteristic of the final consonant. (For illustration, see the third word in the table, which has 3 alternative forms, 2 of them colloquial, and which may exist in 6 pronunciation forms. Let us state that for pronunciation of Czech we use symbols introduced in [12].)

4.1 Creating Structured Lexicon

Before starting the lexicon building process we had to collect some additional text corpora. As mentioned earlier, we already have had access to the three CNC corpora of spoken Czech [6,7]. To be more precise, we were given word unigrams and bigrams derived from them. The two corpora contained about 3 million words in total. The target domain was represented by (non-public) recordings and transcriptions of almost 30 hours of telephone conversations (338 thousand words). From this, a subset of 500 utterances (6,585 words) was kept separate for testing purposes.

As these resources were too small, we searched for data that had similar characteristics of conversation-like informal communication and that were available in significantly larger amounts. On internet we found two sources: a) Czech subtitles to (mostly foreign) movies and b) comments to various forums. The advantage of the former consists in a large amount of transcribed dialogues, the latter covers (often very informal) discussions on diverse topics. The size of the usable data was 25 and 35 million words, respectively.

The lexicon for colloquial speech recognition was built in several iterative steps. In the first one, we used the rules mentioned in Section 3.1 to derive the colloquial forms ("colloquemes") of the words contained in our largest lexicon (the broadcast 350K one). The second level of the lexicon was made of these colloquemes and the standard word forms. In some cases, also non-colloquial spelling variants were added (e.g. "president" and "prezident", or "milion" and "milión"). In the second step, all the available corpora were analyzed to compile lists of OOV words. These were sorted according to their frequency and labeled as either new entries to the lexicon or as colloquemes/variants of the existing ones.

In this way we got the first version of the lexicon. At this moment we could utilize its structure to derive an inverse list that contained mapping of all colloquemes and variants to their normalized form. The list was used to normalize all the corpora, i.e. to replace the alternative word forms by the normalized ones. Because the mapping can be ambiguous in some cases (as demonstrated in Table 4), the list includes a special tag to indicate these cases. When the list is applied to normalize a text corpus, disambiguation is solved by taking into account values in the available bigram LM.

The process of searching for OOV terms, sorting them, adding them to the lexicon and normalizing the corpora was repeated several times. After that, the resulting lexicon contained 386K normalized forms together with additional 37K alternative forms (most of them were colloquemes). In the last step, pronunciations taken from the previous versions of the lexicon or generated by a G2P tool were added.

Table 4. Example of a list of inverse (alternative to normalized form) mappings derived from the lexicon. (The last column contains a binary tag that indicates possible ambiguities.)

Alternative form	Normalized form	Ambiguity
president	prezident	no
malej	malý	no
vod	vod (eng. "water")	yes
vod	od (eng. "from")	yes
sem	sem (eng. "here")	yes
sem	jsem (eng. "I am")	yes

4.2 Language Model Building

For building the language model, four types of corpora were available. Two sources of transcribed oral conversations: CNC corpus (farther denoted as C, with 3M words) and telephone corpus (T, 0.3M words), and the other two sets: subtitle corpus (S, 25M words) and forum corpus (F, 35M words). After normalizing them, we could use them for computing a bigram language model - the default LM in our system.

The lists of word-pairs with their counts were compiled for each corpus separately. As the size of the four corpora differs significantly, these should be mixed in a proper manner to get a balanced LM. Taking the corpus size into account and at the same time trying to avoid the over-fitting effect, we decided to set the following mixing ratios for the initial experiments: T : C : S : F = 10 : 5 : 1 : 1. These numbers were used as the factors to multiply the word-pair counts for each corpus. After mixing the factored pairs and before computing the bigrams, a count normalization step was applied when appropriate. Its goal was to ensure that the smallest count was 1. So, for example, if we decided to mix only the T and C corpora and used the above factors 10 and 5, the resulting counts were divided by 5. As the last step, the bigram LM with Witten-Bell smoothing was calculated.

The following basic LMs were prepared for testing: one based on the oral corpora only (T+C), another derived from the internet resources only (S+F), and one made from all the available data (T+C+S+F).

5 Speech Recognition Experiments

The speech recognition system used in the experiments (also in those mentioned in Section 2.2) operated with the following settings and parameters: telephone signal with 8 kHz sampling frequency, 39 MFCC feature vectors, separate male and female discriminatively trained acoustic models (HMMs), automatic gender recognition via GMMs, and Viterbi decoder optimized for very large vocabularies [10]. The output from the recognizer is a sequence of words from the given lexicon. As default, the words are normalized, but it is possible to get also the recognized alternative forms.

The test data was made of 500 utterances from the telephone conversation set (the same as in Section 2.2). For the purpose of accuracy evaluation, the reference transcriptions were normalized in the same way as the corpora.

In Table 5, we summarize the results achieved with two types of lexicon and 3 types of LM. The large lexicon (386K words) is that described in Section 4.1. The smaller one (110K) is its down-scaled version containing only those words that had frequency higher than 10 in the aggregate corpus (T+C+S+F).

Table 5. Recognition results achieved with different LM sources and different lexicon sizes

Lexicon and LM source	Accuracy [%]	OOV [%]	Time [×RT]
Lex 387K, oral data (T+C)	39.1	1.2	2.9
Lex 387K, internet data (S+F)	34.7	1.2	3.1
Lex. 387K, all data (T+C+S+F)	48.5	1.2	3.0
Lex. 110K, all data (T+C+S+F)	48.2	1.4	1.1

6 Discussion and Conclusions

Table 5 summarizes the most representative results from many experiments we run with different settings and parameters. We were exploring the impact of the mixing factors, the lexicon size, the number of pronunciations, etc. Though the figures were slightly changing in a small range, the positive effect of mixing colloquial and more formal data was always evident. The normalization scheme enabled us to make the lexicon compact and "cleaner" (by putting all the colloquial and alternative forms to the second level). In language model building it helped significantly in dealing with the problem of sparsity because most low-frequency words from the oral corpora got higher bigram counts when normalized and backed up by the large internet sources.

The accuracy rates presented in Table 5 may seem very low, especially when compared to those in Table 1. One should realize, however, that the test set was made of true (not simulated) telephone conversations with spontaneous, sometimes hard to understand, noise-full dialogues. Experiments done with other types of telephone speech, e.g. calls to help lines, or call centrums, yielded significantly better results.

Acknowledgments. The research was supported by the Grant Agency of the Czech Republic (grant no. 102/08/0707).

References

1. Godfrey, J.J., Holliman, E.C., McDaniel, J.: Switchboard: Telephone Speech Corpus for Research and Development. In: Proc. of ICASSP, San Francisco, pp. 517–520 (1992)
2. CALLHOME and CALLFRIEND Corpora in Various Languages. Linguistic Data Consortium, http://www.ldc.upenn.edu/Catalog/
3. Cieri, C., Miller, D., Waller, K.: The Fisher Corpus: A Resource for the Next Generation of Speech-to-Text. In: Proc. of LREC, Lisbon, Portugal, pp. 69–71 (2004)
4. Hain, T., et al.: Automatic Transcription of Conversational Telephone Speech. IEEE Trans. on Speech and Audio Processing 13(6), 1173–1185 (2005)
5. van Leeuwen, D.A., Kessens, J., Sanders, E., van den Heuvel, H.: Results of the N-Best 2008 Dutch Speech Recognition Evaluation. In: Proc. of Interspeech, Brigthon UK, pp. 2571–2574 (2009)
6. Corpus ORAL 2006 and ORAL 2008. Institute of Czech National Corpus. Charles University, Prague, http://www.korpus.cz
7. Corpus PMK. Institute of Czech National Corpus. Charles University, Prague (2001), http://www.korpus.cz
8. Nouza, J., Silovský, J.: Fast Keyword Spotting in Telephone Speech. Radioengineering 18(4), 665–670 (2009)
9. Nouza, J., Žd'ánský, J., Červa, P., Silovský, J.: Challenges in Speech Processing of Slavic Languages (Case Studies in Speech Recognition of Czech and Slovak). In: Esposito, A., Campbell, N., Vogel, C., Hussain, A., Nijholt, A. (eds.) Development of Multimodal Interfaces. LNCS, vol. 5967, pp. 225–241. Springer, Heidelberg (2010)
10. Nouza, J., Žd'ánský, J., Červa, P., Kolorenc, J.: A System for Information Retrieval from Large Records of Czech Spoken Data. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 401–408. Springer, Heidelberg (2006)
11. Schmiedtova, V.: Colloquial Czech in Corpus ORAL 2006 (in Czech). In: Proc. of Conference Czech in Spoken Corpus, Prague, pp. 199–221 (2008)
12. Nouza, J., Psutka, J., Uhliř, J.: Phonetic Alphabet for Speech Recognition of Czech. Radioengineering 6(4), 16–20 (1997)

Fast Phonetic/Lexical Searching in the Archives of the Czech Holocaust Testimonies: Advancing Towards the MALACH Project Visions

Josef Psutka, Jan Švec, Josef V. Psutka, Jan Vaněk, Aleš Pražák, and Luboš Šmídl

Department of Cybernetics, West Bohemia University, Pilsen, Czech Republic
{psutka,svec,psutka_j,vanekyj,aprazak,smidl}@kky.zcu.cz,
http://www.kky.zcu.cz

Abstract. In this paper we describe the system for a fast phonetic/lexical searching in the large archives of the Czech holocaust testimonies. The developed system is the first step to a fulfillment of the MALACH project visions [1,2], at least as for an easier and faster access to the Czech part of the archives. More than one thousand hours of spontaneous, accented and highly emotional speech of Czech holocaust survivors stored at the USC Shoah Foundation Institute as video-interviews were automatically transcribed and phonetically/lexically indexed. Special attention was paid to processing of colloquial words that appear very frequently in the Czech spontaneous speech. The final access to the archives is very fast allowing to detect segments of interviews containing pronounced words, clusters of words presented in pre-defined time intervals, and also words that were not included in the working vocabulary (OOV words).

1 Introduction

The principal goal of MALACH (Multilingual Access to Large Spoken Archives) project was to develop methods for improved access to large multilingual spoken archives collected by the Survivors of the Shoah Visual History Foundation (VHF) between 1994 and 1997. Today these archives of video-interviews are located in the Shoah Foundation Institute at the University of Southern California and contain approximately 52,000 interviews (testimonies) in 32 languages of personal memories of survivors of the World War II Holocaust (116,000 hours of video). More than 550 testimonies of this collection is in Czech (almost 1,000 hours of video). The MALACH project was carried out between 2002–2007 (in cooperation with the VHF, IBM, JHU Baltimore, University of Maryland, CU in Prague and UWB in Pilsen) with the financial support of the NSF. The objective of the MALACH project was to develop and verify techniques for speech recognition of spontaneous, accented and highly emotional speech of holocaust survivors. The plan was to use an output of the recognizer for automatic indexing of pronounced testimonies and automatic searching for keywords and topics. Using multilingual thesaurus this process should work across different languages. Although a great deal of work was done not all objectives were fully fulfilled [3,4]. This paper describes our two years effort to fulfill the MALACH project visions at least for access to the Czech part of archives.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 385–391, 2010.

The state-of-the-art techniques of acoustic and language modeling were applied to build up a LVCSR system which overcomes the former one [5] in recognition accuracy up to 9% absolutely. More than one thousand hours of speech of Czech holocaust survivors stored as video-interviews were then automatically transcribed and phonetically/lexically indexed. Special attention was paid to processing of colloquial words that appear very frequently in the Czech spontaneous speech. The final access to the archives is very fast, allowing to detect segments of interviews containing pronounced words, clusters of words presented in pre-defined time intervals, and also words that were not included in the working vocabulary (OOV words).

2 Characteristics of the Corpora

Testimonies of the Czech holocaust corpus as well as other languages are stored at the Shoah Foundation Institute (SFI) digital library as video interviews. The speech of each interview participant (the interviewer and interviewee) was usually recorded in a quiet rooms via lapel microphones that recorded speech on separate channels. The speech quality in individual interviews is however very poor from the ASR point of view, as it contains whispered or emotional speech with many disfluencies and non-speech events as crying, laughter etc. The speech was also often affected by using many colloquial (non-grammatical) words. The speaking rate (measured as the number of words uttered per minute) varies greatly depending on the speaker (the average age of all speakers was about 75 years), changing from 64 to 173 with the average of 113 [words/minute].

The average length of a Czech testimony is 1.9 hours. Each testimony was divided and stored at SFI as half-hour parts in MPEG-1 video files. For the further processing the audio streams were extracted. The audio track was stored at 128kb/sec stream in 16-bit resolution and 22.05 kHz sampling rate.

For preparing the acoustics 400 speakers were randomly selected. However, only 15 minute segment was transcribed for training purposes per each speaker. This training set contains 42% males and 58% females speakers (it corresponds to the whole database). Another entire 20 testmonies (10 males and 10 females) were transcribed for test a developement sets.

3 Building LVCSR

3.1 Annotation

The audio files were divided into segments and annotated using the special annotation software Transcriber 1.4.1, which is a tool for assisting the creation of speech corpora. Transcriber is freely available from the Linguistic Data Consortium (LDC) web site `http://www.ldc.upenn.edu/` (for details of the annotation process see [2]).

3.2 Acoustic Modeling

The acoustic training portion consisted of 100 hours of Czech speech. The data was parameterized as 15 dimensional PLP cepstral features including their delta and delta-delta derivatives (3 15=45 dimensional feature vectors) [6]. These features were computed at rate of 100 frames per second. Cepstral mean subtraction was applied per speaker. The resulting triphone-based model was trained using HTK Toolkit. The number of clustered states and number of Gaussians mixtures per state was optimized using development test set and had more than 6k states and 16 mixtures per state (almost 100k Gaussians). A silence model was trained by borrowing Gaussians from all non-speech HMMs in proportion to their state and mixture occupancies. The resulting silence model contained 128 mixtures per state and was found to be useful in rejecting non speech events during recognition.

Speaker-adaptive models (SAT) were trained via fMLLR, for each training speaker. After fMLLR transforms for training speakers were computed against the original speaker-independent model, the original model was then re-estimated using the affinely transformed features. This process was repeated few times to converge. The DT model was developed from SAT model via four training iterations based on MMI-FD objective function [7]. Because the speaker identity is available, it can be used to improve the recognition. All training data were split to three clusters (male-speakers female-speakers and interviewer) for DT adaptation. This DT adaptation was done via two iterations DT-MAP on SAT-DT acoustic model [8].

3.3 Language Modeling

Since it is impractical to create enough language model training data by transcribing the speech, we investigated the use of other text collections to complement the transcriptions (see [9] for details). Two basic language models were trained. The first language model was trained on 5.6MB (1.1M tokens) of training set transcriptions. One of the most important issues that had to be decided before the transcription process started is the way of transcription of colloquial words. As explained in details in [4] the best performance of ASR is obtained by using the colloquial forms during acoustic model training while restricting the language model to the formal forms both in the lexicon and in the LM estimation process.

The second language model was trained from the selection of the Czech National Corpus (CNC). This corpus is relatively large (approximately 400M words) and is extremely diverse. Therefore we investigated the possibility of using automatic methods to select sentences from the CNC that are similar in language usage, lexicon and style to the sentences in the training set transcriptions. This in-domain selection from CNC contains 82MB of text (16M tokens). An interpolated language model has been created with the ratio 2:1 (transcriptions to the CNC). The resulting trigram language model with modified Kneser-Ney smoothing contains 252k words (308k phonetical variants). Language models were estimated using the SRI Language Modeling Toolkit (SRILM) [10].

3.4 Word and Phoneme Lattices

Due to the stereo speech signal (the interviewer on one channel and interviewee on the other) and to the conversation character of the data the special algorithm was introduced in order to reduce the huge amount of speech data. Only parts, where at least one speaker was talking, were further processed. In the case, where interviewer and interviewee were cross-talking, both channels were recognized separately. This task was quite challenging because there occurred echoes even though the speakers had lapel microphones. During recording, the speech of interviewer and interviewee mixed together so that each speaker was recorded in both channels, only with different level of energy.

The LVCSR system was designed to work in two passes. In the first pass, clustered DT adapted acoustic models was automatically adapted to each of 550 speakers, using a bigram language model. This automatic iterative fMLLR+MAP adaptation [11] used only speech segments with posterior probabilities over 0.99. Word lattices were then generated based on information about word transitions performed during the second pass recognition by LVCSR system. The lattices were built up retroactively, from the last recognized word by adding other most probable word hypotheses (alternatives) of the recognized words according to the desired depth (number of concurrent hypotheses) of word lattices. For searching through word lattices, the posterior probability computed by forward-backward algorithm was assigned to each hypothesis in the word lattice. Normalized acoustic likelihoods and a trigram language model were used during the lattices computation. Due to the effect of the segmentation of the word graph [12], posterior probabilities for different hypotheses of the same word were summed.

Phoneme lattices were generated in the same manner, based on information about phoneme transitions performed during the recognition by phoneme recognizer without use of any language model. This recognizer was built for each speaker on its acoustic model adapted by the first pass of LVCSR system.

The parameters for the LVCSR system were optimized on the development data (whole testimonies of 5 male and 5 female speakers). The recognition results depicted in the Table 1 show the phoneme recognition accuracy as well as recognition accuracy for LVCSR system. This results were enumerated on the test set (another 5 male's and 5 female's testimonies). The total number of words in the test set was 63,205 with 2.39% out-of-vocabulary (OOV) words.

Table 1. The results of recognition experiment

Recognition level	Acc [%]
LVCSR	71.44
Phonemes	70.38

4 Indexing and Searching

4.1 Indexing

To achieve a very good responsiveness of the searching system, we decided to create an index of both the word lattices and the phonetic lattices. The index was created using the SQL database that was able to store the huge number of records. During the indexing procedure the structural properties of lattices were omitted from the database and only the word or phone occurrences were stored.

Word lattice index was relatively simple. Every lattice arc represented one word and the word with corresponding time was added into the index if the score of the arc exceeded some threshold. The index of phonetic lattice was more complicated because indexing single phones was not effective – the database query was not very specific and search would return many unrelated results. Therefore we decided to index trigrams of the subsequent phones. The trigrams are overlapped and can be simply generated from the phonetic lattice by virtue of using the breadth-first search. The score of the trigram was obtained as an average score of the three phones in the trigram. Again, only trigrams with the score larger then some threshold (different from the word lattice threshold) were added into the index.

In total, the database contains about 100M records and there is about 27 records for every second of the indexed audio signal. The index contains 63,329 unique trigrams of phones. For instance, the five most frequent are: `sem` (140,248 occurrences), `bil` (125,977), `tak` (120,972), `ese` (117,515) and `oto` (117,252).

4.2 Searching

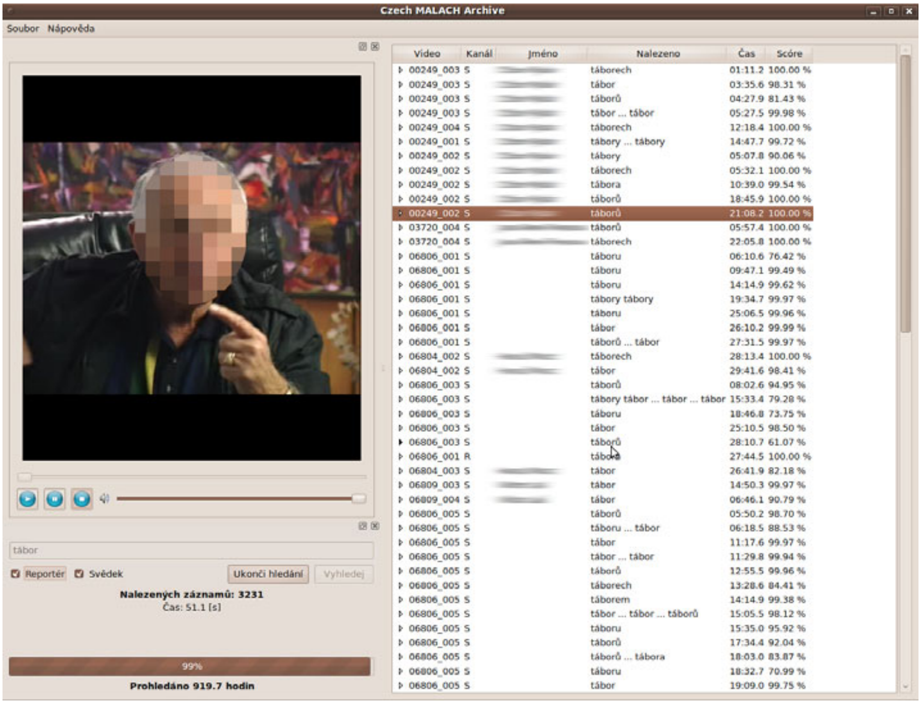
Unlike the laboratory (off-line) system for full-scale IR such as [13] this system searches only keywords and/or key-phrases but is extremely fast and fully interactive. The searching algorithm depends on the searched word. First, a lemma of the searched word is generated. Then if the lemma is found in the vocabulary the word lattice search is performed. It is also possible to search for all possible forms (from the lemma) of the searched vocabulary word. This behavior can be optionally disabled if the user wants to search only the exact word form.

If the searched word is an out-of-vocabulary word the phonetic lattice search is performed. The phonetic transcription of the word is generated and the overlapping trigrams of phones are computed. Then the database query selects the records corresponding to these trigrams. The results are grouped by a corresponding audio track and ordered by the time. If the searched word occurs in the audio track at the given time there must be a cluster of the trigrams. The algorithm we use does not strictly require the presence of all trigrams from the searched word. The score of the word occurrence is computed from the score of indexed trigrams and the total number of found trigrams.

The time required to search through the whole archive strongly depends on the searched word itself, mainly on the number of occurrences. For in-vocabulary words the time needed to search the whole archive is typically between 5 and 10 seconds. The out-of-vocabulary words are typically searched between 30 and 60 seconds.

5 GUI Description

The graphical user interface is designed to be as simple as possible. We suppose that the users of our searching tool will be mostly non-technicians. The user enters the searched word into the text box and selects the channel or channels to be searched (the reporter and/or the witness). The user can also enter a phrase instead of a single word. There are also used some operators in the search engine to modify the number of results: to tag the word as required (it must occur in every result) the user should write the plus sign in



tábor

☑ Reportér ☑ Svědek Ukonči hledání Vyhledej

Nalezených záznamů: 3231
Čas: 51.1 [s]

99%
Prohledáno 919.7 hodin

Video	Kanál	Jméno	Nalezeno	Čas	Score
00249_003	S	[redacted]	táborech	01:11.2	100.00 %
00249_003	S	[redacted]	tábor	03:35.6	98.31 %
00249_003	S	[redacted]	táborů	04:27.9	81.43 %
00249_003	S	[redacted]	tábor ... tábor	05:27.5	99.98 %
00249_004	S	[redacted]	táborech	12:18.4	100.00 %
00249_001	S	[redacted]	tábory ... tábory	14:47.7	99.72 %
00249_002	S	[redacted]	tábory	05:07.8	90.06 %
00249_002	S	[redacted]	táborech	05:32.1	100.00 %
00249_002	S	[redacted]	tábora	10:39.0	99.54 %
00249_002	S	[redacted]	táborů	18:45.9	100.00 %
00249_002	S	[redacted]	táborů	21:08.2	100.00 %
03720_004	S	[redacted]	táborů	05:57.4	100.00 %
03720_004	S	[redacted]	táborech	22:05.8	100.00 %
06806_001	S		táboru	06:10.6	76.42 %
06806_001	S		táboru	09:47.1	99.49 %
06806_001	S		táboru	14:14.9	99.62 %
06806_001	S		tábory tábory	19:34.7	99.97 %
06806_001	S		táboru	25:06.5	99.96 %
06806_001	S		tábor	26:10.2	99.99 %
06806_001	S		táborů ... tábor	27:31.5	99.97 %
06804_002	S	[redacted]	táborech	28:13.4	100.00 %
06804_002	S	[redacted]	tábor	29:41.6	98.41 %
06806_003	S		táborů	08:02.6	94.95 %
06806_003	S		tábory tábor ... tábor ... tábor	15:33.4	79.28 %
06806_003	S		táboru	18:46.8	73.75 %
06806_003	S		tábor	25:10.5	98.50 %
06806_003	S		táborů	28:10.7	61.07 %
06806_001	R		tábo[illegible]	27:44.5	100.00 %
06804_003	S	[redacted]	tábor	26:41.9	82.18 %
06809_003	S	[redacted]	tábor	14:50.3	99.97 %
06809_004	S	[redacted]	tábor	06:46.1	90.79 %
06806_005	S		táborů	05:50.2	98.70 %
06806_005	S		táboru ... tábor	06:18.5	88.53 %
06806_005	S		tábor	11:17.6	99.97 %
06806_005	S		tábor ... tábor	11:29.8	99.94 %
06806_005	S		táborů	12:55.5	99.96 %
06806_005	S		táborech	13:28.6	84.41 %
06806_005	S		táborem	14:14.9	99.38 %
06806_005	S		tábor ... tábor ... táborů	15:05.5	98.12 %
06806_005	S		táboru	15:35.0	95.92 %
06806_005	S		táborů	17:34.4	92.04 %
06806_005	S		táborů ... tábora	18:03.0	83.87 %
06806_005	S		táboru	18:32.7	70.99 %
06806_005	S		tábor	19:09.0	99.75 %

Fig. 1. Searching of the word *"tábor"*

front of it and to find the exact form of the word it should be enclosed into parenthesis. For example the searching of the word *"tábor"* (*camp*) can be seen in Figure 1.

6 Conclusion

This paper presents the system for a fast phonetic/lexical searching in the large archives of the Czech holocaust testimonies. Nearly 1,000 hours of interviews are searched typically up to 10 seconds for in-vocabulary words and up to 60 seconds for out-of-vocabulary words. Without this tool searching for a specific event or situation was complicated, since this huge amount of data had to be searched manually. This searching tool made the interviews more accessible to the historians, students, researchers, and actually to the whole public. By now is the Czech part of the holocaust survivors database stored at the SFI, the only one with such searching engine.

Acknowledgements

This research was supported by the Ministry of Education of the Czech Republic, projects No. MSM235200004 and No. LN00A063 and by the grant of The University of West Bohemia, project No. SGS-2010-054.

References

1. Byrne, W., Doerman, D., Franz, M., Gustman, S., Hajič, J., Oard, D., Picheny, M., Psutka, J., Ramabhadran, B., Soergel, D., Ward, T., Zhu, W.: Automatic Recognition of Spontaneous Speech for Access to Multilingual Oral History Archives. IEEE Transactions on Speech and Audio Processing 4, 420–435 (2004)
2. Psutka, J., Ircing, P., Psutka, J.V., Radová, V., Byrne, W., Hajič, J., Gustman, S., Ramabhadran, B.: Automatic Transcription of Czech Language Oral History in the MALACH Project: Resources and Initial Experiments. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2002. LNCS (LNAI), vol. 2448, pp. 253–260. Springer, Heidelberg (2002)
3. Psutka, J., Ircing, P., Psutka, J.V., Hajič, J., Byrne, W., Mírovský, J.: Automatic Transcription of Czech, Russian and Slovak Spontaneous Speech in the MALACH Project. In: Interspeech Lisboa 2005, pp. 1349–1352. ISCA, Bonn (2005)
4. Psutka, J., Ircing, P., Hajič, J., Radová, V., Psutka, J., Byrne, W., Gustman, S.: Issues in Annotation of the Czech Spontaneous Speech Corpus in the MALACH Project. In: Fourth International Conference on Language Resources and Evaluation, pp. 607–610. European Language Resources Association, Lisbon (2004)
5. Psutka, J., Hajič, J., Byrne, W.: The Development of ASR for Slavic Languages in the MALACH Project. In: Acoustics, Speech, and Signal Processing, pp. 749–752. IEEE, Piscataway (2004)
6. Hermansky, H.: Perceptual Linear Predictive (PLP) Analysis of Speech. J. Acoustic. Soc. Am. 87 (1990)
7. Povey D.: Discriminative Training for Large Vocabulary Speech Recognition. Ph.D. thesis, Cambridge University, Department of Engineering (2003)
8. Vaněk, J., Psutka, J.V., Zelinka, J., Pražák, A., Psutka, J.: Discriminative Training of Gender-Dependent Acoustic Models. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS (LNAI), vol. 5729, pp. 331–338. Springer, Heidelberg (2009)
9. Psutka, J., Ircing, P., Psutka, J.V., Radová, V., Byrne, W., Hajič, J., Mírovský, J., Gustman, S.: Large Vocabulary ASR for Spontaneous Czech in the MALACH Project. In: EUROSPEECH 2003 Proceedings, pp. 1821–1824. ISCA, Geneva (2003)
10. Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit. In: International Conference on Spoken Language Processing (ICSLP 2002), Denver, USA (2002)
11. Zajíc, Z., Machlica, L., Müller, L.: Refinement Approach for Adaptation Based on Combination of MAP and fMLLR. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 274–281. Springer, Heidelberg (2009)
12. Wessel, F., Schlüter, R., Macherey, K., Ney, H.: Confidence Measures for Large Vocabulary Continuous Speech Recognition. IEEE Transactions on Speech and Audio Processing 9(3) (2001)
13. Ircing, P., Müller, L.: Benefit of Proper Language Processing for Czech Speech Retrieval in the CL-SR Task at CLEF 2006. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 759–765. Springer, Heidelberg (2007)

CORPRES

Corpus of Russian Professionally Read Speech

Pavel Skrelin, Nina Volskaya, Daniil Kocharov,
Karina Evgrafova, Olga Glotova, and Vera Evdokimova

Department of Phonetics, Saint-Petersburg State University
Universitetskaya Emb., 11, 199034, Saint-Petersburg, Russia
{skrelin,volni,kocharov,evgrafova,oglotova,postmaster}@phonetics.pu.ru
http://www.phonetics.pu.ru

Abstract. The paper introduces CORPRES - COrpus of Russian Professionally REad Speech developed at the Department of Phonetics, Saint Petersburg State University, as a result of a three-year project. The corpus includes samples of different speaking styles produced by 4 male and 4 female speakers. Six levels of annotation cover all phonetic and prosodic information about the recorded speech data, including labels for pitch marks, phonetic events, phonetic, orthographic and prosodic transcription. Precise phonetic transcription of the data provides an especially valuable resource for both research and development purposes. Overall corpus size is 60 hours of speech. The paper contains information about CORPRES design and annotation principles, and overall data description. Also, we discuss possible use of the corpus in phonetic research and speech technology as well as some findings on the Russian sound system obtained from the corpus data.

Keywords: Phonetics, speech corpus, text-to-speech, annotation, phonetic transcription, manual transcription, prosodic feature labelling.

1 Introduction

Contemporary research both in linguistic phonetics and speech technology is largely based on and can benefit from the use of large speech corpora. The corpus to be used for these purposes must contain a large sample of speech data, ensure a consistently high quality of data, and it must also have annotation that enables researchers of a wide range of phonetic issues to search for and to find specific data that is valid and reliable. Good examples of such a resource are the corpora developed for Dutch [1] and Russian [2]. The need for a fully annotated large corpus of Russian speech recorded at a consistently high quality is evident.

In this paper we present CORPRES, a fully annotated COrpus of Russian Professionally REad Speech developed at the Department of Phonetics, Saint Petersburg State University, as a result of a three-year project. The corpus meets all of the requirements for databases of this kind which are listed above and it may be used for the purposes of both application development and scientific research. It is large enough for statistical machine learning and has six annotation levels including prosodic annotation, rule-based and manual phonetic transcription reflecting the actual sounds pronounced by the

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 392–399, 2010.

speakers. Manual expert segmentation of 40% of the corpus and expert annotation also make the corpus an excellent database for phonetic research of contemporary Russian.

2 Corpus Design

The corpus was originally intended for use in unit-selection TTS synthesis. However, it appeared expedient to create a corpus that would be a good representation of Standard Russian speech suitable as a basis for a wider range of research, e.g. variation and change in Standard Russian, among other topics. This goal predetermined the following principles for the corpus design.

Primarily, the corpus was intended as a sample of Standard Russian (St. Petersburg pronunciation variant); dialect variation was not accounted for. Secondly, the corpus was to represent a number of speaking styles, both more deliberate and resembling spontaneous speech. Thirdly, the corpus was to be phonetically and prosodically rich, i.e. to contain a large number of all Russian phonemes in all possible contexts and a wide range of diverse prosodic structures, as well as to provide for good lexical representation. In order to be suitable for variability research, it had to contain realizations of the same language units by a number of speakers, both men and women. Fourthly, it was necessary to ensure a consistent high quality of all data both in terms of technical characteristics and voice quality. The final, and the most crucial objective we had in mind was to ensure that the annotation of the corpus covers a wide range of information that may be of interest to those involved in most areas of phonetic research.

These principles resulted in the following corpus design. The corpus includes only read speech in order to ensure that the same linguistic units (phonemes, words, prosodic structures) are used by more than one speaker. Different styles of texts were selected for recording with specific characteristics of those styles in mind:

Text A – an action-oriented fiction narrative resembling conversational speech in sentence length and word choice;

Text B – a fiction narrative of a more descriptive nature containing longer sentences and very little direct speech;

Text C – a play containing a high number of conversational remarks and emotionally expressive dialogues and monologues;

Text D – purely informational neutral text on IT;

Text E – purely informational neutral text on politics and economy issues. Both informational texts contain terminology, geographical and proper names, numerals, acronyms and abbreviations.

Records were made from eight speakers, four men and four women, in order to cover a certain degree of variation within the St. Petersburg pronunciation. All of them were professional speakers: some of them worked in radio broadcasting; some were professional actors or television newsmen. Speaker selection was based on a number of criteria such as pleasantness of voice and clear articulation.

Each speaker read three texts. All eight speakers read both fiction narrative texts (A and B). The choice of the third text was based on the speaker's reading manner and background. Speakers with acting experience (two women and two men) read the play

(C). Speakers with newscasting experience read one of the informational texts: one man and one woman read the text on IT (D), while the other man and woman read the text on politics and economy issues (E).

The recordings were made in the recording studio at the Department of Phonetics. Motu Traveler multi-channel recording system, an AKG capacitor microphone and WaveLab software were used. The recordings have a sample rate of 22,050 Hz and a bit rate of 16 bits. Before the recording sessions, all texts were revised to detect and resolve ambiguities caused by nonstandard words, terminology etc. In case of doubt, speakers could ask for instructions from researchers present at the studio. Slips of the tongue were noted, and the parts where they occurred were recorded again.

Overall corpus size is 528,458 running words and contains 60 hours of speech made up of 7.5 hours from each speaker. The corpus was annotated by a team of experts on six annotation levels that will be further discussed in greater detail. 40% of the corpus (24 hours of speech) was manually segmented and fully annotated on all six levels. 60% of the corpus was partly annotated; there are labels for pitch period and phonetic event labels. Orthographic and prosodic transcription, as well as rule-based phonetic transcription (see Section 3 for detail) for this part was generated and then stored as text files, but was not transferred to sound file labels. The fully annotated part of the corpus covers all speaking styles included in the corpus and all speakers. Table 1 shows general corpus statistics. The number of phonemes in the part of the corpus which was not annotated on phonetic transcription levels has not been estimated, therefore, two cells in the table remain empty.

Table 1. General corpus statistics

	Fully annotated data	Partly annotated data	Total amount
Phonemes	1,048,867	–	–
Words	211,437	317,021	528,458
Tone units	64,055	86,546	150,601
Hours	24	36	60

3 Annotation

The annotation captures the maximum amount of phonetically and prosodically relevant data. The six annotation levels are as follows:

Level 1 – pitch marks;

Level 2 – phonetic events labelling;

Level 3 – manual phonetic transcription (the sounds actually pronounced by the speakers);

Level 4 – rule-based phonetic transcription (automatically generated by a text transcriber);

Level 5 – orthographic transcription;

Level 6 – prosodic transcription.

Levels 1 and 2 contain information on various phonetic events: epenthetic vowels, voice onset, voiced plosure, stationary parts of voiceless consonants, laryngalization, and glottalization. Phonetic events were annotated manually by expert phoneticians. Level 5 also contains information on perceptually prominent words. Prosodic transcription in Level 6 includes labels for different types of pauses, types of tone unit, and non-speech events such as laughter or breathing. Orthographic and prosodic transcription labels, phonetic transcription labels of word-initial allophones, and fundamental frequency labels in case of voiced signal were automatically aligned with each other.

3.1 Detecting and Labelling Periods of Fundamental Frequency

Fundamental frequency periods were detected automatically. A linear combination of the following methods was used for this purpose: autocorrelation, analysis-by-synthesis, spectral domain analysis, estimation of the energy of signal peaks and the correlation of neighboring periods. The efficiency of automatic pitch detection and pitch period labeling was about 98%. The results of the automatic procedure were checked and corrected manually.

3.2 Phonetic Transcription

In CORPRES, transcription is available on two levels. Level 3 contains narrow manual phonetic transcription which reflects the sounds actually pronounced by the speakers. The rule-based transcription found at Level 4 was generated in accordance with a set of phonological rules without reference to the actual sound. The rules were developed in accordance with the transcription tradition of the St. Petersburg Phonological School. Thus, the rules yielded a phonetic transcription that is normally used for isolated words as they appear in pronunciation dictionaries, with some account of word-boundary assimilation processes. As a result, Level 4 contains a conventional phonetic transcription of the speech sample. The transcription symbols used were a version of SAMPA for the Russian language. To mark positional allophones of 6 Russian vowel phonemes /a/, /o/, /i/, /u/, /e/, /y/, 18 symbols were used. Each vowel symbol contained indication of the sound's position regarding stress. Thus, 0 was used for a stressed accented vowel, 1 – for an unstressed vowel in a pretonic syllable, 4 – an unstressed one in a post-tonic syllable. The set of consonant symbols included 41 symbols to cover 36 Russian consonant phonemes and 5 voiced allophones of voiceless consonants which frequently occur at word junctions.

To produce the manual phonetic transcription, the speech signal was segmented, transcribed and peer-revised by expert phoneticians. Rule-based phonetic transcription was generated automatically by a text transcriber. The labels were automatically aligned with the label positions produced manually on the manual transcription level. The results of this procedure were not perfect due to the mismatch of rule-based and manual phonetic transcriptions, and therefore, they were further manually corrected.

3.3 Orthographic and Prosodic Transcription

Prosodic information was marked by expert phoneticians on the basis of perceptual and acoustic analysis of the speech data in a text file containing orthographic transcription.

The labels were later automatically transferred from the text file to the annotation files to coincide with the phonetic transcription levels. Orthographic transcription was stored on Level 5; it contains the boundaries of words and word labels. Besides, the perceptually prominent words are labeled with special symbols. Prosodic information was stored on Level 6; this level contains the boundaries of tone units and pauses and their labels. The set of symbols to label pauses and tone units and the principles behind the labeling process are described in detail in [3].

4 The Use of the Corpus

As CORPRES contains a large sample of high quality speech data with detailed annotation, it enables researchers of a wide range of phonetic issues to search for and find specific data that is valid and reliable. Thus, the corpus may be useful in a wide range of phonetic research. The necessary information from the corpus (e.g. sound variants and their frequency distribution etc.) is obtained by means of computer software specially designed to suit a certain task. For instance, we are able to obtain information on vowel variation. Table 2 below shows the ratio between vowel realizations according to the rule-based transcription (down) and the manual transcription (across). 0 is used to mark a stressed vowel, 1 – a pretonic vowel, and 4 – a post-tonic vowel. The column Total shows the total number of corresponding allophones.

This data shows that there is a certain degree of variation even for stressed vowels that tend to be more stable than the unstressed ones, with approximately 1–3% of them pronounced as allophones of other phonemes. Some of the unstressed vowels are especially unstable, e.g. less than 50% of post-tonic /a/ vowels are pronounced as /a/, while a third of them are pronounced as /y/ allophones. The vowel variation findings support those obtained earlier on a smaller corpus of read and spontaneous speech [4].

A closer look at vowel variation data provides insight into the changes in Standard Russian. The general phonotactic rule for unstressed vowels is that /e/ and /o/ do not generally occur in the unstressed position, but can be found in a small number of words, mostly loan words and foreign names, and contexts (post-tonic /e/ and /o/ are mostly found in word-final open syllables) (e.g. *radio /r a0 d’ i4* ***o4****, izvinite /i1 z v’ i1 n’ i0 t’* ***e4****/, Hemingway /h* ***e1*** *m’ i1 n g u1 e0 j/*). Our data showed that unstressed /e/ is pronounced as /i/ or /y/ in about 40% of the cases. The unstressed /o/ is pronounced in 92.3% and appears to be very stable. Therefore, we may assume that the phonotactics of Standard Russian is going through change in this respect.

Some interesting observations can be also made on the subphonemic level. A good example of this is the case of the vowel insertion or epenthetic vowels. This vowel sound emerges in certain consonant clusters, but does not enjoy a phonological status. The epenthetic vowel within a consonant cluster may trigger the appearance of an additional syllable and thus change the rhythmic structure of the word. Therefore, a closer look at this phenomenon is useful for considering both segmental and suprasegmental characteristics of Russian sound structure, as well as for the issues of pronunciation dictionary design and initial transcription alignment in ASR systems and production of

Table 2. Rule-based vs. manual transcription: vowels

	a	e	i	o	u	y	Total
a0	98.3	1.5		0.1		0.1	52,769
a1	80.7	3.9	0.1	1.6	0.5	13.1	76,992
a4	46.3	13.2	1.6	4.6	1.3	33.0	53,667
e0		97.6	1.0	0.4		0.9	30,861
e1	0.6	61.0	13.2	0.6	0.6	23.9	159
e4		55.6	18.9	1.1	2.2	22.2	90
i0		0.5	98.9		0.1	0.5	20,596
i1	0.1	6.2	91.0	0.2	0.8	1.8	47,840
i4	0.6	19.0	77.4	0.3	0.9	1.9	38,799
o0	0.1	0.2		99.1	0.2	0.3	43,875
o1	1.3	0.3	0.1	93.4	2.2	2.8	1,945
o4	7.1	3.0		71.7	5.1	13.1	99
u0				0.2	99.7	0.1	12,503
u1			0.2	0.9	98.5	0.4	12,729
u4	0.2	1.6	0.9	2.4	92.8	2.1	9,144
y0		0.4	0.6		1.0	97.9	9,355
y1	1.3	6.9	7.1	0.8	2.0	81.9	6,275
y4	1.0	9.2	0.3	0.8	2.0	86.7	14,337

more natural speech in TTS. In the corpus, epenthetic vowels were labeled and marked with an asterisk (*). Table 3 below shows the total amount of consonant junctions in the corpus and the amount of clusters containing epenthetic vowels.

Table 3. Epenthetic vowels in CORPRES

	Epenthetic Vowels	Consonant Junctions
Count	24,785	206,781
Percentage	12 %	100 %

The corpus contains 24,785 instances of epenthetic vowels, which is a sufficient amount of data for extensive research into the nature of this phenomenon. A small amount of the corpus data was used in an earlier experiment described in [5]. As a result, the acoustic characteristics of epenthetic vowels were obtained: it is a mid-central vowel in a non-palatalized consonant cluster and a close fronted vowel in palatalized contexts. It was also determined that epenthetic vowels are most common in clusters containing a consonant and either /r/ or /l/.

Rule-based transcription does not completely reflect the actual sound. We compared the total amount of phonemes in the rule-based transcription reflecting the way the speech sample is supposed to be pronounced according to the conventional transcription rules to that in the manual transcription reflecting the way it actually was pronounced by the speakers recorded. Table 4 shows whether the expectations based on the rule-based transcription were met in the actual recording in percentage terms. Table 4 reveals that despite the fact that as much as 84.7% of the rule-based transcription reflects the actual pronunciation, 9.05% of the expected sounds are replaced by other sounds, and 6.25 % of the expected sounds are actually not pronounced at all. Thus, the manual transcription does not match 15.3% of the rule-based transcription.

Table 4. Rule-based vs. manual transcription

	Total	Match	Mismatch	Elisions
Count	1,118,833	947,508	101,292	70,033
Percentage	100%	84.7%	9.05%	6.25%

As the annotated part of the corpus used for this analysis includes an even distribution of all of the represented speaking styles and speakers, we can expect that similar results could be obtained from the analysis of the rest of the corpus. This clearly shows that the rule-based transcription alone does not yield data that would be sufficient or valid for any type of phonetic research or practical application. Therefore, despite the large amount of human and financial resources required, precise phonetic transcription seems to be an indispensable part of corpus annotation at the present moment. There appear to be two ways of overcoming the discrepancy between rule-based transcription and manual transcription. One possible solution is to bring the automatic transcriber up-to-date by using the obtained information about the actual sound pronunciation. In this respect, the present corpus and its two levels of phonetic transcription may be used as a database for revising the traditional view of Standard Russian pronunciation and introducing new phonetic transcription rules. The other solution is to avoid automatic rule-based transcription altogether and transcribe all of the data manually. The former course of action appears to be more preferable as the emergence of a set of rules reflecting the current state of the language would largely benefit both the development of speech technology applications and theoretical research in Russian phonetics.

5 Conclusion

We developed a fully-annotated large corpus of Russian speech including samples of different speaking styles produced by eight professional speakers: four men and four women. Overall corpus size is 528,458 running words and contains 60 hours of speech, made up of 7.5 hours from each speaker. The six levels of annotation cover all phonetic and prosodic information about the recorded speech data. Forty percent of the corpus (24 hours of speech) was fully annotated on all six levels; the remaining sixty percent of the corpus was partly annotated; there are labels for pitch period and phonetic event labels. Orthographic and prosodic transcription, as well as the rule-based phonetic transcription is stored as text files, but was not aligned with the sound files.

Precise phonetic transcription of the data provides an especially valuable resource for both research and development. The corpus may be used for unit-selection TTS synthesis purposes, as well as a bootstrapping corpus for speech recognition systems, or as data for research in Russian phonetics and inter- and intra-speaker variability.

Acknowledgment

The research reported on in this paper was supported in part by the "Research Potential Development of Higher Education Institutions" Federal Program ("Automatic Interpretation of Prosody" Project).

References

1. Van Son, R.J.J.H., Binnenpoorte, D., Van Den Heuvel, H., Pols, L.C.W.: The IFA Corpus: a Phonemically Segmented Dutch "Open Source" Speech Database. In: Proceedings of the 7th European Conference on Speech Communication and Technology, Aalborg, pp. 2051–2054 (2001)
2. Bondarko, L.V., Volskaya, N.B., Tananaiko, S.O., Vasilieva, L.A.: Phonetic Properties of Russian Spontaneous Speech. In: Proceedings of the 15th International Congress of Phonetic Sciences, Barselona, pp. 2977–2980 (2003)
3. Volskaya, N.B., Skrelin, P.A.: Prosodic Model for Russian. In: Proceedings of Nordic Prosody X, Peter Lager, Frankfurt am Main (2009), pp. 249–260 (2009)
4. Bolotova, O.: On Some Acoustic Features of Spontaneous Speech and Reading in Russian (Quantitative and Qualitative Comparison Methods). In: Proceedings of the 15th International Congress of Phonetic Sciences, Barcelona, pp. 913–916 (2003)
5. Evgrafova, K.V.: Sravnenie foneticheskih harakteristik glasnoj vstavki v chtenii izolirovannyh slov i svyaznogo teksta (na material russkogo jazyka). In: Formalnye metody analiza russkoj rechi: Materialy sektsii XXXVIII Mezhdunarodnoj filologicheskoj konferencii, St. Petersburg, pp. 3–10 (2009) (in Russian)

Hybrid HMM/BLSTM-RNN for Robust Speech Recognition

Yang Sun, Louis ten Bosch, and Lou Boves

Department of Linguistics, Radboud University, Nijmegen, The Netherlands
{y.sun,l.tenbosch,l.boves}@let.ru.nl
http://lands.let.ru.nl

Abstract. The question how to integrate information from different sources in speech decoding is still only partially solved (layered architecture versus integrated search). We investigate the optimal integration of information from Artificial Neural Nets in a speech decoding scheme based on a Dynamic Bayesian Network for noise robust ASR. A HMM implemented by the DBN cooperates with a novel Recurrent Neural Network (BLSTM-RNN), which exploits long-range context information to predict a phoneme for each MFCC frame. When using the identity of the most likely phoneme as a direct observation, such a hybrid system has proved to improve noise robustness. In this paper, we use the complete BLSTM-RNN output which is presented to the DBN as Virtual Evidence. This allows the hybrid system to use information about all phoneme candidates, which was not possible in previous experiments. Our approach improved word accuracy on the *Aurora 2* Corpus by 8%.

Keywords: automatic speech recognition, noise robustness, hybrid HMM/RNN, virtual evidence, dynamic Bayesian network.

1 Introduction

Speech recognition performance degrades dramatically under noise. Many techniques have been developed by modifying different steps of the whole recognition process, such as speech enhancement [1], feature extraction [2], and speech modeling [3]. Nevertheless, there is still a large performance gap between Automatic Speech Recognition (ASR) and Human Speech Recognition (HSR).

While Hidden Markov Models (HMM) are the dominant statistical approach to automatic speech recognition, Recurrent Neural Networks (RNN) have also shown excellent performance as discriminative classifiers. Moreover, it has been shown that hybrid HMM/RNN architectures can make for very powerful and efficient speech recognition systems [4]. In order to improve noise robustness, [5] proposed an architecture that integrates a novel Bidirectional Long Short-term Memory Recurrent Neural Network (BLSTM-RNN) in a HMM system implemented as a graphical model. To limit the computational complexity, only the index of the most likely phoneme of the RNN output was used as direct observation. This approach was promising since BLSTM-RNN is able to exploit long range temporal dependencies from both input directions (forward and backward), which enhances noise robustness.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 400–407, 2010.

However, by only using the identity of the most probable phoneme as an additional direct observation, a large part of the information provided by the BLSTM-RNN is ignored. In addition, it may well be that the estimate of the most probable phoneme is not correct. Figure 1 shows the outputs of the BLSTM-RNN for an isolated digit *ONE* for different SNR levels. It is easy to see that for clean speech the output is crisp and correct. Both the body of a phoneme and the start and end points are unambiguous (and correct). However, when SNR decreases the situation becomes less clear and it may well be that some of the winning phoneme estimates are wrong. Yet, during these intervals the network may still provide useful evidence in favour of the correct phoneme. Therefore, in this paper we investigate whether integrating the whole output probability vector of the BLSTM-RNN as Virtual Evidence (VE) in the DBN can improve recognition performance. Treating the BLSTM-RNN output as Virtual Evidence allows to integrate a distribution over its domain instead of a single 'observed' value. A complete distribution over all phoneme candidates may prevent the correct estimate from being eliminated from the input, as was the case in [5].

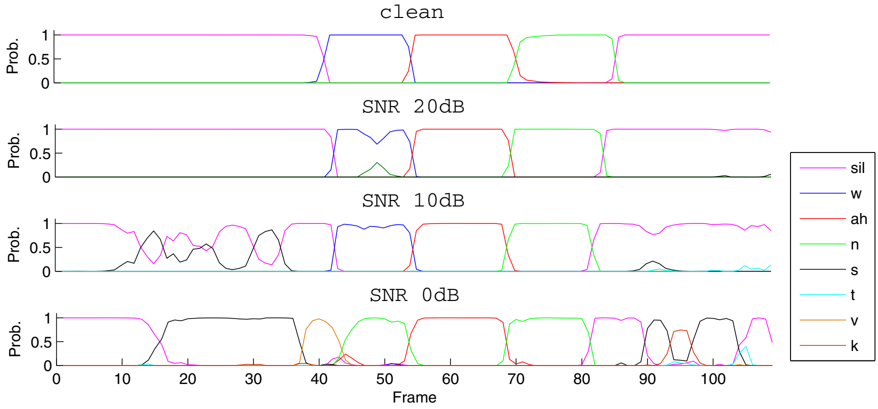


Fig. 1. BLSTM-RNN output predictions for single utterance *ONE* in different SNR cases. Curves blur dramatically as the noise energy increases, especially phoneme w (*blue curve*) almost disappears in SNR 0dB.

The structure of the paper is as follows: Section 2 briefly introduces the BLSTM-RNN. Section 3 and Section 4 describe VE and how it is integrated to build a hybrid HMM/RNN system, respectively. Finally we discuss our experimental results in Section 5 and make the conclusion in Section 6.

2 BLSTM-RNN

Although in principle RNNs can account for very wide contexts by allowing feedback from a large number of previous inputs, in practice realistic context windows become quite limited due to the so-called vanishing gradient problem [6]. This led to the introduction of Long Short-term Memory RNN (LSTM-RNN) [7]. The structure of LSTM-RNN is basically the same as a classic RNN, but now each hidden neuron is replaced by a so-called LSTM memory block. Input gates correspond to a read operation, which allows inputs to pass while the gate is open. Output gates perform analogous to a write operation, allowing outputs to flow to connected nodes. Forget gates act as a reset button, clearing the memory when they are opened. In this architecture, old inputs are well preserved and accessible for processing of far later outputs. In numerous pattern recognition tasks [8,10,11,12], LSTM-RNNs have shown excellent performance.

Another drawback of a traditional RNN is that it can only access past context, while in [13] it was shown that knowledge of the future is equally important as knowledge of the past. Similar to the design of Bidirectional RNNs, [15] added a parallel chain with a backward direction in the LSTM-RNN hidden layer. Thus, this novel RNN can use context information both from forward and backward directions. This architecture is called Bidirectional LSTM-RNN (BLSTM-RNN). Properly trained with acoustic features, this network architecture can provide noise robustness by exploiting a long temporal range context.

3 Virtual Evidence

In addition to directly observing acoustic parameter o_t at the time t and using a conditional probability table (CPT) or Gaussian Mixtures to assign a likelihood $p(s_j|o_t)$ to state s_j when o_t is the observation feature, it is also possible to integrate a discrete probability distribution via so-called *Virtual Evidence* (VE) in a graphical model. VE is used to provide a "prior distribution" over all the candidates. The use of VE substantially increases the poser of DBNs, by providing the ability of using probabilistic knowledge from external sources. For example, in our case, the posterior likelihood $p(s_j|o_t)$, which is the output activation from an independent BLSTM-RNN system, is regarded as a prior probability that is observed indirectly by DBN. Since BLSTM-RNNs or other discriminative classifiers can be much more accurate than a Maximum Likelihood classifier (e.g. Gaussian mixture model), we expect that the integration of a BLSTM-RNN as Virtual Evidence will enhance the performance of a DBN that receives only direct observations as input.

The VE sub-structure in a graphical model is depicted as the gray nodes in Figure 2. The corresponding factorization is:

$$p(\mathbf{s}, \mathbf{o}, \mathbf{v}) \propto \prod_t p(v_t = 1|o_t)p(o_t|s_t) \tag{1}$$

where $p(v_t = 1|o_t) \propto p(s_j|o_t)$ which is obtained from external systems. This probability distribution is then read in at each frame rather than calculated. For more details, see [17] and [18].

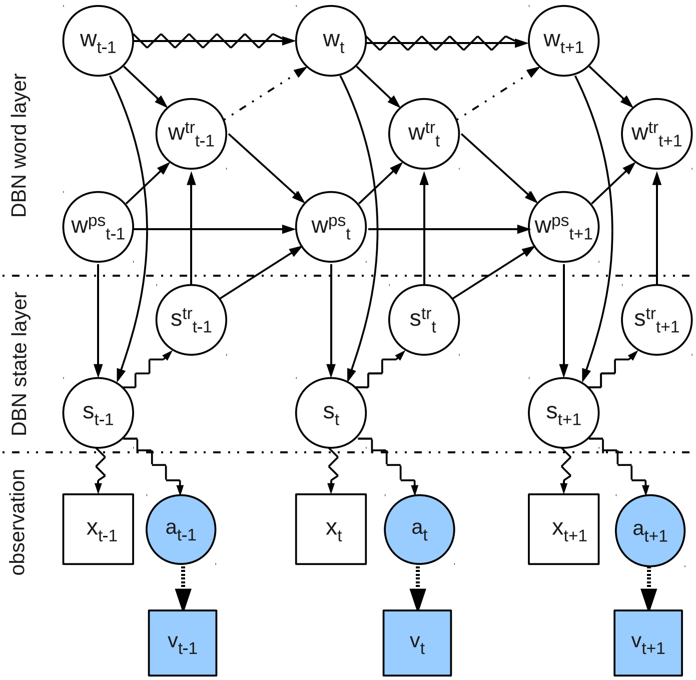


Fig. 2. Architecture of the hybrid HMM/RNN

4 DBN Architectures

In this study, we built 5 DBN architectures in total for a systematical evaluation.

- DBN only observes MFCC(M),
- DBN only observes BLSTM as an index of the most likely phoneme B(I),
- DBN only observes BLSTM, but treats them as virtual evidence B(V),
- Tandem DBNs observe both MFCC and BLSTM outputs as I and V, named M/B(I) and M/B(V) respectively.

Figure 2 depicts the tandem architecture with MFCC and BLSTM outputs as virtual evidence. In the figure circles represent hidden variables in this architecture and observed variables are represented by squares. Shaded nodes represent VE provided by the BLSTM-RNN. Node a_t is the actual state and node v_t is the virtual one, which is always set to 1. Straight lines indicate deterministic conditional probability functions (CPFs), random CPFs correspond to zig-zag lines; dotted lines correspond to switching parent dependency.

In the top word layer, node w_t models the word. w_{tr} indicates whether a word transition occurs or not. w_{ps} is the position within the current word. $w_{ps} = 1 : S$, where S is the total number of states of the current word. In the state layer, all the states s_t are represented frame-wisely. Node s_{tr} is a state transition variable. We designed it such that, when $w_{ps} = S$ and a state transition happens ($s_{tr} = 1$), then a word transition is forced to take place. Finally in the observation layer, x_t indicates the acoustic features. a_t and v_t comprise the virtual evidence sub-structure described in Section 3.

Thanks to the flexibility of the DBN, the basic structures for all the 5 DBNs are exactly the same in our experiments. The only differences occur in the observation layer – 5 combinations of observation MFCC(M), BLSTM as index(B(I)) and BLSTM as VE(B(V)).

5 Results and Discussion

The experiments presented in this paper were conducted on the *Aurora 2* database [19], which consists of recognizing sequences of digits contaminated by different noise types. Since we aim to investigate the optimal way of integrating information from a RNN, the model is only trained by clean speech and tested on the test set A with different SNR levels of four noise types (subway, babble, car noise and exhibition hall).

Table 1. Word accuracies on *Aurora 2* set A

Noise Type	SNR	M	B(I)	B(V)	M/B(I)	M/B(V)	HTK
Subway	0 dB	13.17%	27.69%	**35.46%**	23.12%	22.26%	27.30%
	10 dB	69.67%	74.84%	80.63%	84.10%	**85.78%**	78.72%
	20 dB	97.79%	92.72%	94.66%	96.99%	**98.13%**	96.96%
	clean	**99.32%**	98.50%	98.50%	99.08%	99.26%	98.83%
Babble	0 dB	-5.05%	20.24%	**27.48%**	15.48%	14.30%	11.73%
	10 dB	41.05%	78.70%	**81.32%**	59.95%	67.44%	49.06%
	20 dB	84.37%	96.33%	**96.74%**	92.81%	94.86%	89.96%
	clean	**99.67%**	98.97%	99.03%	99.49%	99.61%	98.97%
Car	0 dB	11.00%	24.36%	**30.53%**	15.53%	17.05%	13.27%
	10 dB	49.79%	77.02%	**80.53%**	67.23%	75.07%	66.24%
	20 dB	91.14%	93.73%	94.36%	94.84%	**95.59%**	96.84%
	clean	97.41%	97.02%	96.99%	97.41%	**97.44%**	98.81%
Exhibition	0 dB	18.14%	31.23%	**37.37%**	21.29%	28.29%	15.98%
	10 dB	74.95%	70.93%	78.19%	82.29%	**86.39%**	75.10%
	20 dB	97.22%	91.96%	94.91%	96.70%	**97.84%**	96.20%
	clean	98.95%	98.49%	98.40%	98.86%	98.95%	**99.14%**
	mean	64.91%	73.30%	**76.97%**	71.57%	73.64%	69.57%

12 cepstral mean normalized *MFCC* features together with energy as well as first and second order delta coefficients were extracted from the speech signal (same as used in the baseline experiments [19]). These acoustic features were used as the input of BLSTM-RNN to compute a posterior probability for each phoneme. The size of this RNN's input layer equals 39 (dimension of *MFCC* features) while the size of its

output layer is 20 (19 different phonemes occurring in the *Aurora* digit strings plus one silence). Both forward and backward hidden layers contain 100 memory blocks of one cell each. We did a supervised training of the network on a forced aligned frame-wise phoneme transcriptions and its output activations were considered as external BLSTM-RNN features.

Similar to the baseline recognizer in [19], our DBN consists of 16 states per digit, a silence model of 3 states and a short pause model shares the middle state of the silence model. All the 39 dimensions of *MFCC* features are represented by Gaussian Mixtures, which were split once 0.02% convergence was reached. The final model has up to 32 Gaussian Mixtures. The BLSTM-RNN output was integrated into the DBN either as an extra direct discrete observation by using the index of the most likely phoneme (I) as in [5], or via VE (V) to incorporate the whole probability vector. Table 1 presents the performance (word accuracy in percentage) in various conditions.

From Table 1 a remarkable improvement (over 3% on average) can be seen from integrating the complete probability vector as Virtual Evidence compared to treating the output of the BLSTM-RNN as a discrete directly observed feature. More importantly, the benefit from B(V) over B(I) increases as the SNR decreases. This result can be explained by the fact that the impact of incorrect ‘winners’ in the VE approach is less detrimental than in the original architecture where the output of the BLSTM-RNN was always treated as ‘true’. To explain this, we refer to the example in Figure 1, where it can be seen that the output of the BLSTM-RNN is corrupted in low SNR conditions. Many false predictions show up, some even with a high probability score. Obviously, in these cases, reducing the BLSTM-RNN outputs to one single discrete index highly reduces the chance that the true phoneme is saved for input in the DBN. On the other hand, since the BLSTM-RNN makes few errors with clean test data, the advantage of using VE is not significant for clean speech.

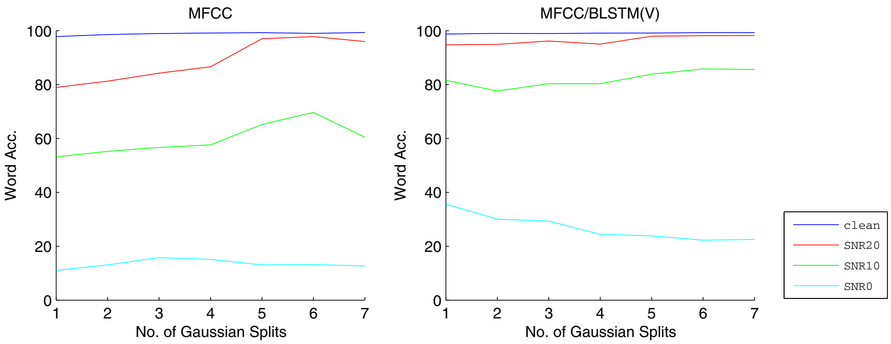


Fig. 3. Results of validation tests during training MFCC and MFCC/BLSTM-RNN models. All the Gaussian Mixtures are split once 0.02% convergence was reached. After each split, the model was trained around 15 iterations until it is convergent again.

The results of the tandem architecture, combining MFCC and BLSTM-RNN features, again show the advantage of using BLSTM features as VE. The fifth column

(M/B(V)) outperforms the fourth (M/B(I)) by 3%. However, this tandem model M/B(V) does not always provide a better result than single BLSTM features. More specifically, tandem model M/B(V) is inclined to perform better than B(V) in high SNR cases. Figure 3 shows the validation results during the training progress for M and M/B(V). In the training progress of M, all of the testing results improved performance when the number of Gaussian splits increases, except SNR0 which stayed relatively stable over the number of Gaussian splits.

6 Conclusion

In this work, we focused on how to integrate information from an external discriminative classifier (BLSTM-RNN) into a DBN. Different from [5], where the index of the most likely phoneme of RNN outputs is regarded as a direct observation, the whole output probability vector is incorporated as virtual evidence.

We showed that the use of the full BLSTM-RNN output gives significantly better results than using only the best phoneme index, in particular for low SNRs. The use of the tandem architecture again shows advantages of the use of the entire output vector from the BLSTM-RNN.

As a next step, a new training strategy will be studied to resolve performance decay during training of the Tandem model in low SNR cases. In addition, other types of virtual evidence will be introduced to DBN as virtual evidence, for example using a support vector machine for phoneme classification. Finally, we will investigate virtual evidence predicting states or words, rather than phonemes.

Acknowledgments

We acknowledge Jort Florent Gemmeke for useful discussion. The research of Yang Sun is funded through the Marie Curie Initial Training Network SCALE.

References

1. Lathoud, G., Magimia-Doss, M., Mesot, B., Boulard, H.: Unsupervised Spectral Subtraction for Noise-Robust ASR. In: Proc. of ASRU, San Juan (2005)
2. Droppo, J., Acero, A.: Noise Robust Speech Recognition with a Switching Linear Dynamic Model. In: Proc. of ICASSP, Montreal (2004)
3. Mesot, B., Barber, D.: Switching Linear Dynamic Systems for Noise Robust Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing 15(6), 1850–1858 (2007)
4. Bourlard, H., Morgan, N.: Connectionist Speech Recognition: A Hybrid Approach. Kluwer Academic Publishers, Dordrecht (1994)
5. Wöllmer, M., Eyben, F., Schuller, B., Sun, Y., Moosmayr, T., Nguyen-Thien, N.: Robust In-Car Spelling Recognition – a Tandem BLSTM-HMM Approach. In: Proc. of Interspeech, Brighton (2009)
6. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies. In: Kremer, S.C., Kolen, J.F. (eds.) A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, Los Alamitos (2001)

7. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation 9(8), 1735–1780 (1997)
8. Fernandez, S., Graves, A., Schmidhuber, J.: An Application of Recurrent Neural Networks to Discriminative Keyword Spotting. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 220–229. Springer, Heidelberg (2007)
9. Schuster, M., Paliwal, K.: Bidirectional Recurrent Neural Networks. IEEE Transactions on Signal Processing 45, 2673–2681 (1997)
10. Wöllmer, M., Eyben, F., Keshet, J., Graves, A., Schuller, B., Rigoll, G.: Robust Discriminative Keyword Spotting for Emotionally Colored Spontaneous Speech using Bidirectional LSTM Networks. In: Proc. of ICASSP, Taipei (2009)
11. Graves, A., Fernandez, S., Liwicki, M., Bunke, H., Schmidhuber, J.: Unconstrained Online Handwriting Recognition with Recurrent Neural Networks. Advances in Neural Information Processing Systems (2008)
12. Wöllmer, M., Eyben, F., Reiter, S., Schuller, B., Cox, C., Douglas-Cowie, E., Cowie, R.: Abandoning Emotion Classes – Towards Continuous Emotion Recognition with Modelling of Long-Range Dependencies. In: Proc. of Interspeech, Brisbane, pp. 597–600 (2008)
13. Graves, A.: Supervised Sequence Labelling with Recurrent Neural Networks. Ph.D. thesis (2008)
14. Rigoll, G., Neukirchen, C.: A New Approach to Hybrid HMM/ANN Speech Recognition Using Mutual Information Neural Networks. In: Advances in Neural Information Processing Systems, (NIPS 1996), vol. 9, pp. 772–778 (2008)
15. Graves, A., Fernandez, S., Schmidhuber, J.: Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 602–610. Springer, Heidelberg (2005)
16. Morgan, N., Bourlard, H.: An Introduction to Hybrid HMM/Connectionist Continuous Speech Recognition. IEEE Signal Processing Magazine, 25–42 (May 1995)
17. Bilmes, J.: On Soft Evidence in Bayesian Networks. Technical Report UWEETR-2004-0016, University of Washington, Dept. of EE (2004)
18. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Francisco (1988)
19. Hirsch, G. H., Pearce, D.: The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions. In: ISCA ITRW ASR 2000: Automatic Speech Recognition: Challenges for the Next Millennium, Paris (2000)

Some Aspects of ASR Transcription Based Unsupervised Speaker Adaptation for HMM Speech Synthesis

Bálint Tóth, Tibor Fegyó, and Géza Németh

Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
{toth.b,fegyo,nemeth}@tmit.bme.hu
http://www.tmit.bme.hu/speechlab

Abstract. Statistical parametric synthesis offers numerous techniques to create new voices. Speaker adaptation is one of the most exciting ones. However, it still requires high quality audio data with low signal to noise ration and precise labeling. This paper presents an automatic speech recognition based unsupervised adaptation method for Hidden Markov Model (HMM) speech synthesis and its quality evaluation. The adaptation technique automatically controls the number of phone mismatches. The evaluation involves eight different HMM voices, including supervised and unsupervised speaker adaptation. The effects of segmentation and linguistic labeling errors in adaptation data are also investigated. The results show that unsupervised adaptation can contribute to speeding up the creation of new HMM voices with comparable quality to supervised adaptation.

Keywords: HMM-based speech synthesis, unsupervised adaptation, automatic speech recognition.

1 Introduction

In the last decade the primary goal of speech synthesis was to achieve natural sounding, high quality voices. As the results of unit selection and statistical parametric speech synthesis improve, new challenges emerge. Creating a new voice, which is similar to the voice characteristics of a target speaker, is an attractive challenge.

Context independent unit selection synthesis demands a well constructed speech database with hours of speech, its phonetic transcription and precise labeling for each new voice. This method is time consuming and a lot of human interaction is necessary. Statistical parametric synthesis offers speaker adaptation techniques, where a speech database of moderate size is required only to create a similar voice to the target speaker's. Human interaction is still necessary for precise phonetic transcription and labeling.

As the quality of statistical parametric speech synthesis approaches the quality of state-of-the-art unit selection methods it became a focused research area. Usually the HMM paradigm—well known from the speech recognition domain—is used in statistical speech synthesis [1]. It has numerous advantages compared to unit selection: small footprint, the possibility of creating various voices [2], emotional speech [3] and

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 408–415, 2010.

adapting the voice characteristics to a target speaker [4,5]. Recently hybrid approaches, like target cost prediction of unit selection systems by HMMs [6], smoothing the segment sequence of unit selection systems with statistical models and/or their dynamic features [7], mixing unit selection and statistical parametric speech synthesis [8] have also been proposed.

2 Supervised and Unsupervised Adaptation

In HMM speech synthesis and recognition the two main techniques of speaker adaptation are maximum likelihood linear regression (MLLR) [4] and maximum a posteriori (MAP) estimation [5]. MLLR is applied when the amount of adaptation data is small, for MAP more data is required as the Gaussian distributions are updated individually. In both cases supervised speaker adaptation uses precise phonetic transcriptions, manually transcribed or automatically annotated segmentation and linguistic labels.

The advantages of unsupervised adaptation of HMM speech synthesis are quite appealing - the creation of target voices becomes automatic which is favorable if several voices are required or if pre-processing of the speech data is not possible. Probably the most advanced method would be to create a full-context speech recognizer and train the HMMs with the output of this system. Although no studies have been carried out, it is likely to be computationally inadequate and would probably create inaccurate labels.

In Automatic Speech Recognition (ASR) systems both supervised and unsupervised adaptation are used to increase the recognition accuracy. The unsupervised method requires less manual work, but more adaptation data; about one hour per speaker is used in practice [9].

In [10] an interesting method of unsupervised speaker adaptation was introduced. In this study only phonetic labels were used for adaptation, the transformation matrices were computed from triphone models. The results of the study show that the degradation in quality and naturalness is caused mainly by limiting full-context labels to triphone labels, and not by triphone mismatches.

Another study [11] investigates a two-pass decision tree construction technique for unsupervised adaptation. The decision trees of full context models are built in two phases: first the segmental, then the supra-segmental features are processed. According to the results of [11] there is no perceived quality difference between supervised and unsupervised adaptation, although the average voice was trained by ASR corpora, so it produces very low quality synthetic speech (1.9–2.0 MOS values [11]), which may hide the quality degradation caused by this two-pass method.

Another important aspect is described in [12]. Several tests of different TTS systems with the same labels and clear and noisy speech database are carried out. The results of [12] show that HMM-based adaptive speech synthesis is far more robust than concatenative, speaker-dependent HMM-based, or hybrid speech synthesis approaches.

3 ASR-Based Unsupervised Speaker Adaptation

Complementing the results of [9,10,11,12] our concept is to evaluate the quality of adaptation with inaccurate, noisy phonetic transcription. The consequences of

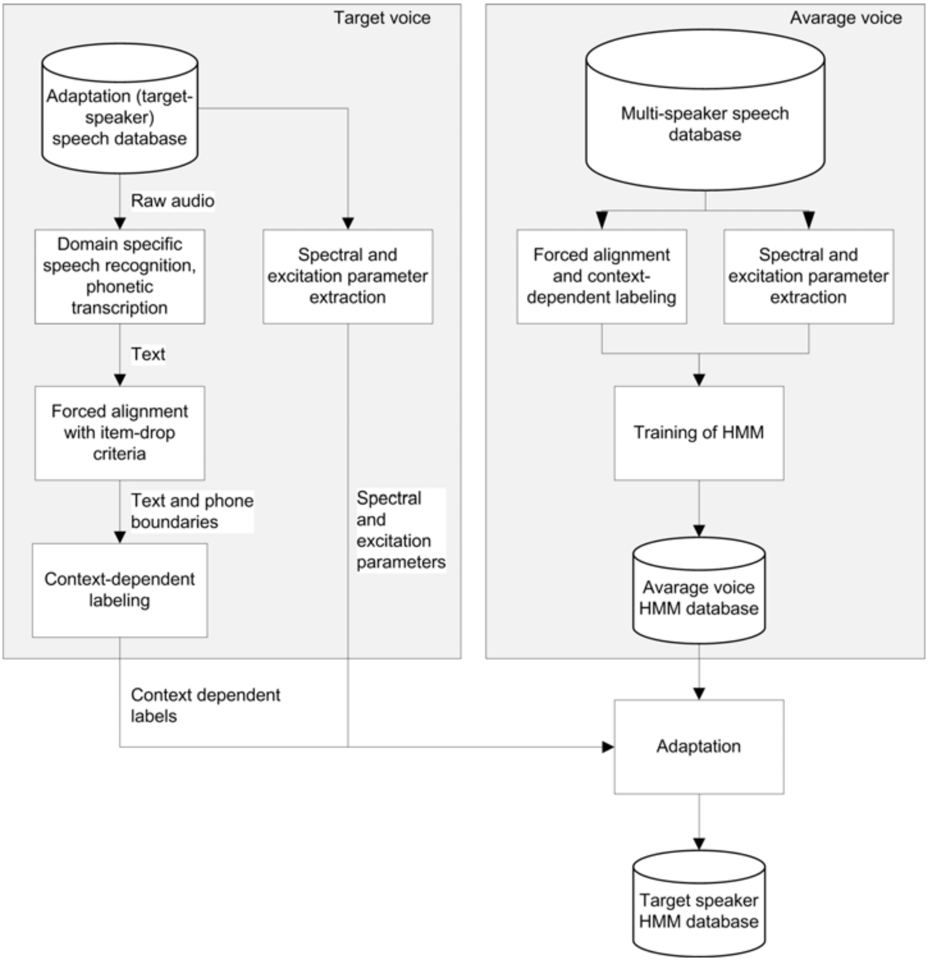

Target voice
Adaptation (target-speaker) speech database
Raw audio
Domain specific speech recognition, phonetic transcription
Spectral and excitation parameter extraction
Text
Forced alignment with item-drop criteria
Text and phone boundaries
Context-dependent labeling
Spectral and excitation parameters
Avarage voice
Multi-speaker speech database
Forced alignment and context-dependent labeling
Spectral and excitation parameter extraction
Training of HMM
Avarage voice HMM database
Context dependent labels
Adaptation
Target speaker HMM database

Fig. 1. Block diagram of the proposed unsupervised adaptation method

inaccurate phonetic transcription are phoneme mismatches, inaccurate segmentation and linguistic labels due to phoneme mismatch accumulation.

Speech recognizers for a given context perform quite well, but their output still contains various mismatches.

3.1 The Proposed Method

The speech recordings from the target speaker are recognized, then phone boundaries are determined with forced alignment based on the recognition results. If the results of forced alignment do not satisfy an item-drop criterion (which is described in 3.3) that part of the recordings is rejected. When phone boundary detection is accepted for at least ten minutes of recordings, linguistic labeling is carried out. Finally the adaptation is applied. The block diagram of the proposed method is shown in Fig. 1.

3.2 Automatic Recognition of the Speech Corpus and Phonetic Transcription

The TTS adaptation database is transcribed automatically with an LVCSR ASR system [9]. The output will contain recognition errors, which can be significantly reduced if the context of the TTS adaptation database and the ASR training database are from the same domain. The following processing step is transforming the orthographic output of the ASR system into phonetic representation. This may be completed either by dictionary or rule-based software modules.

3.3 Phone Boundary Detection

The phone boundaries in the TTS adaptation database are marked automatically based on the phonetic transcription described in Section 3.2 using the ASR system in forced alignment mode enabling a narrow beam only. As the word level ASR can produce recognition errors, the length of the recognized phone sequence is likely to be longer or shorter than the correct transcription. If at the beginning of an audio segment the word is misrecognized with more/less phones compared to the correct word then the forced alignment procedure probably gives bad results for the whole audio segment. If this happens at the end of an audio segment, it is not so severe because it will produce only some phone mismatches. To avoid using adaptation data with critical phone error accumulation, the following drop criterion was introduced:

$$e_{accumulation} = 1 - \frac{1}{\prod_{i=1}^{i_{max}} (i_{max} - i + 1)^{(100-p_{ci})*\frac{i*100}{i_{max}}}} <= \epsilon \qquad (1)$$

where i is the position of the phone, i_{max} is the length of the phone sequence, p_{ci} is the confidence, that the i-th phone is correctly recognized in the [0..100] interval (which is computed by the ASR) and ϵ is the limit of the drop criteria in the [0..1] interval (0 means there were no errors, 1 is the theoretical worst case). So mistakes at the beginning are more weighted than at the end and error accumulation is avoided.

4 Results

To measure the difference between the proposed method and the supervised adaptation technique a listening test was conducted. In the experiment a modified Hungarian version of HTS [13] was used. The average voice was computed from five speakers (1.5–2 hours of phonetically balanced speech corpus from each). The adaptation database contained semi-spontaneous (parliament speeches by politicians), 10 minute long speech from each of four different speakers. For adaptation the Constrained Maximum Likelihood Linear Regression (CMLLR) method was used.

For speech recognition a state-of-the-art Hungarian LVCSR system was applied [14]. The triphone based acoustic model was trained with 5 hours of speech from 500 speakers. The training corpus of the morpheme trigram language model contained 1.2 million words in the domain of political news. The average accuracy of the system is 72%, while the average phone accuracy is above 85%. For the TTS adaptation database the accuracy of the recognizer in phonetic level is shown in Table 1.

Table 1. Accuracy of the recognizer for the four speakers

Speaker	Phone accuracy
Speaker #1	58%
Speaker #2	79%
Speaker #3	87%
Speaker #4	90%

In case of supervised speaker adaptation consensus manual phonetic transcription with punctuation was created, the segmentation and linguistic labels were automatically determined. In case of unsupervised adaptation the phonetic transcription was determined from the recognition results, the segmentation and linguistic labels were determined in the same way, as in case of supervised speaker adaptation.

In the test the supervised and unsupervised adaptation from all speakers -altogether eight systems- were involved.

4.1 Experimental Conditions

The experiment consisted of three main parts: paired comparison, Mean Opinion Score (MOS) test and naturalness evaluation. In the first section test subjects had to define how similar two synthesized samples are on a five point scale. The text of the utterance in one pair was always the same. Altogether 24 pairs were played: 8 pairs were from the same system; 8 pairs came from the same speaker with different adaptation methods; and 8 pairs were compiled from different speakers. Pair comparison as the first part is beneficial, because test subjects get used to the synthetic voice and they will give consistent answers for the MOS test of the second part. There the test subjects had to mark the quality of 32 samples, 4 samples from each system. In the last section test subjects had to decide how much the synthesized samples are similar to the natural voice of the original speaker. This was carried out with 40 synthesized samples (5 for each system).

The order of the three parts is chosen in this way to minimize the chance that the test subjects memorize the speakers.

The samples were selected from a large set in order to get the desired information about the systems and not about the speech samples. In every section the synthesized samples were pseudo-randomly selected from the larger sample database keeping the distribution of samples and eight different systems even. The authors carried out a pre-test with four subjects to verify the effectiveness of the test design. The results of the pre-test were promising, consequently the same design was kept.

Altogether 25 test subjects (19 male, 6 female) were involved in the test. The test was internet-based, the average age was 35, and the youngest subject was 21, the oldest 67 years old. 10 test subjects were speech experts.

4.2 Analysis of Results

Table 2 shows the results of the experiment. The first three columns (Similarity to synthesized voice) are related to the first section of the test, the fourth column (Similarity to native voice, same speaker) is related to the third section of the test, and the last column (MOS) is related to the second part of the test. The *s* rows correspond to supervised adaptation, while *u* rows refer to unsupervised adaptation. In the first and third test sections 1 refers to the lowest, 5 to the highest similarity. In the MOS test 1 is the worst, 5 is the best value. Except column three higher values represent better results for all speakers.

Table 2. Results of the listening test (s: supervised, u: unsupervised)

		Similarity to				**MOS**
		Synthetized voice			**Native voice same speaker**	
		Same speaker		**different speaker**		
		s	u			
Speaker #1	s	4.8	4	1.7	2.8	3
	u	4	4.7	2	2.6	2.7
Speaker #2	s	4.8	4.2	1.9	2.9	3.2
	u	4.2	4.6	2.1	2.4	3
Speaker #3	s	4.7	4.4	1.8	2.7	3
	u	4.4	4.9	2.1	2.8	3
Speaker #4	s	4.7	4.4	1.9	2.6	3.1
	u	4.4	4.7	2	2.7	3.1
Standard deviation	s	0.47–0.54	0.62–0.75	0.64–0.75	1.07–1.18	1.05–1.08
	u	0.62–0.75	0.43–0.58			
Confidence ($\alpha = 0.05$)	s u	0.19–0.21	0.18–0.21	0.28–0.32	0.20–0.22	0.21–0.23
Test section		1.			3.	2.

Individual analysis of the results. The first two columns show that test subjects can tell, if the samples were generated from the same speaker with the same methods (*s*-*s*, *u*-*u* samples). There is a minor impact of using different adaptation methods: *s*-*u*, *u*-*s* samples score consequently less than *s*-*s*, *u*-*u* pairs. The third column shows that in case of these four speakers the subjects could tell, if the synthesized samples are from different speakers. Based on the values of the fourth column, both supervised and unsupervised samples are considered moderately similar to the native speakers, but they are still scored much better, than different speakers. The relative low values can be the result of the adaptation data being semi-spontaneous speech, including sputter, echo, cough and hesitation. This is also the reason for rather low MOS scores, which are shown in the fifth column. The standard deviation- and confidence level intervals ($\alpha = 0.05$) are also shown in Table 2.

Analyzing the trends of the results. Each part of the test shows, that the difference between supervised and unsupervised adaptation reduces as the phone accuracy of the ASR system (see Table 1) gets higher. This trend can be seen by examining the following pairs:

- *s*-*s*, *u*-*u* samples compared to *s*-*u*, *u*-*s* samples from the same speaker,
- the *u* and *s* samples similarity of speaker #1,2,3,4 to a different speaker,
- the *u* and *s* samples similarity of speaker #1,2,3,4 to the native voice of the same speaker,
- the MOS scores of *s* and *u* samples.

The results show that the proposed unsupervised adaptation method with good phone accuracy produced similar quality to supervised adaptation with semi-spontaneous adaptation data. Creating new HMM voices can be speeded up by the proposed method. Phone accuracy as low as 58% may still allow with unsupervised adaptation the creation of a comparable voice to the supervised one.

5 Conclusions

In this paper a method for unsupervised adaptation of HMM-based speech synthesis systems was introduced and the quality evaluation of the technique was investigated. As the results are quite promising further studies will be carried out. The parameters of the drop criteria (described in 3.3) will be fine-tuned and other types of drop criteria will be investigated. Unsupervised minimum generation error linear regression (MGELR) and constrained structural maximum a posteriori linear regression (CSMAPLR) adaptation methods will be evaluated. Listening tests will be carried out using the adaptation data presented in this paper and with studio quality data as well.

Acknowledgments. This research was supported by the TELEAUTO (OM-00102/2007) project of the Hungarian National Office for Research and Technology and by the ETOCOM project (TAMOP-4.2.2-08/1/KMR-2008-0007) through the Hungarian National Development Agency in the framework of the Social Renewal Operative Programme supported by EU and co-financed by the European Social Fund and by the KMOP-1.1.3-08/A- 2009-0006 project through the Hungarian National Development Agency.

References

1. Black, A., Zen, H., Tokuda, K.: Statistical Parametric Speech Synthesis. In: ICASSP 2007, pp. 1229–1232 (2007)
2. Iwahashi, N., Sagisaka, Y.: Speech Spectrum Conversion Based on Speaker Interpolation and Multi-Functional Representation with Weighting by Radial Basis Function Networks. Speech Communications 16(2), 139–151 (1995)

3. Tachibana, M., Yamagishi, J., Masuko, T., Kobayashi, T.: Speech Synthesis with Various Emotional Expressions and Speaking Styles by Style Interpolation and Morphing. IEICE Trans. Inf. Syst. E88-D(11), 2484–2491 (2005)
4. Tamura, M., Masuko, T., Tokuda, K., Kobayashi, T.: Adaptation of Pitch and Spectrum for HMM-Based Speech Synthesis Using MLLR. In: ICASSP 2001, pp. 805–808 (1998)
5. Ogata, K., Tachibana, M., Yamagishi, J., Kobayashi, T.: Acoustic Model Training Based on Linear Transformation and MAP Modification for HSMM-Based Speech Synthesis. In: ICSLP 2006, pp. 1328–1331 (2006)
6. Kawai, H., Toda, T., Ni, J., Tsuzaki, M., Tokuda, K.: XIMERA: A New TTS from ATR Based on Corpus-Based Technologies. In: ISCA SSW5 2004, pp. 179–184 (2004)
7. Plumpe, M., Acero, A., Hon, H.-W., Huang, X.-D.: HMM-Based Smoothing for Concatenative Speech Synthesis. In: ICSLP 1998, pp. 2751–2754 (1998)
8. Okubo, T., Mochizuki, R., Kobayashi, T.: Hybrid Voice Conversion of Unit Selection and Generation using Prosody Dependent HMM. IEICE Trans. Inf. Syst. E89-D(11), 2775–2782 (2006)
9. Mihajlik, P., Fegyó, T., Tüske Z., Ircing, P.: A Morpho-graphemic Approach for the Recognition of Spontaneous Speech in Agglutinative Languages like Hungarian. In: Interspeech 2007, pp. 1497–1500 (2007)
10. King, S., Tokuda, K., Zen, H., Yamagishi, J.: Unsupervised Adaptation for HMM-Based Speech Synthesis. In: Interspeech 2008, pp. 1869–1872 (2008)
11. Gibson, M.: Two-Pass Decision Tree Construction for Unsupervised Adaptation of HMM-Based Synthesis Models. In: Interspeech 2009, pp. 1791–1794 (2009)
12. Yamagishi, J., Ling, Z., King, S.: Robustness of HMM-Based Speech Synthesis. In: Interspeech 2008, pp. 581–584 (2008)
13. Tóth, B., Németh, G.: Hidden Markov Model Based Speech Synthesis System in Hungarian. Infocommunications Journal LXIII(2008/7), 30–34 (2008)
14. Mihajlik, P., Tarján, B., Tüske, Z., Fegyó, T.: Investigation of Morph-based Speech Recognition Improvements across Speech Genres In: Interspeech 2009, pp. 2687–2690 (2009)

Online TV Captioning of Czech Parliamentary Sessions*

Jan Trmal[1], Aleš Pražák[2], Zdeněk Loose[1], and Josef Psutka[1]

[1] Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic
{jtrmal,zloose,psutka}@kky.zcu.cz
[2] SpeechTech, s.r.o, Plzen, Czech Republic
ales.prazak@speechtech.cz

Abstract. In the paper we introduce the on-line captioning system developed by our teams and used by the Czech Television (CTV), the public service broadcaster in the Czech Republic.

The research project is targeted at incorporation of speech technologies into the CTV environment. One of the key missions is the development of captioning system supporting captioning of a "live" acoustic track. It can be either the real audio stream or the audio stream produced by a shadow speaker. Another key mission is to develop software tools and techniques usable for training the shadow speakers.

During the initial phases of the project we concluded that the broadcasting of the Parliamentary meetings of the Chamber of Deputies fulfills the necessary conditions that enable it to be captioned without the aid of the shadow speaker. We developed a fully automatic captioning pilot system making the broadcasting of Parliamentary meetings of the Chamber of Deputies accessible to the hearing impaired viewers.

The pilot run enabled us and our partners in the Czech TV to develop and evaluate the complete captioning infrastructure and collect, review and possibly implement opinions and suggestions of the targeted audience.

This paper presents our experience gathered during first years of the project to the public audience.

1 Introduction

The first application of ASR (Automatic Speech Recognition) system for real-time captioning of live broadcasts has arguably been announced by BBC in 2003 [1]. Since then, similar systems have been developed and employed in production use in several countries all around the world (see for example [2,3,4]). However, the speech recognition done on real world acoustic track is much more difficult than the recognition accuracies presented in vast majority of the scientific papers would suggest – consider speaker dialects and speaker emotional states, necessity of very large vocabularies for highly inflectional languages, diverse acoustic environments and large variety of signal distortions introduced by different technical equipments used for acoustic signal transport, storage and processing. To overcome majority of such problems,

* This research was supported by the Ministry of Education of the Czech Republic, project No. 2C06020 and by the University of West Bohemia, project No. SGS-2010-054.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 416–422, 2010.

an alternative approach, called "shadow speaker" (or re-speak) approach is often employed. The principle of such approach is as follows. Instead of generating of the captions from the real-world acoustic track, an indirect approach is used. The real-world track is listened to and re-spoken by a skilled and specifically trained employee – the shadow speaker.

This simplifies the task of ASR significantly – the shadow speaker works in a quiet environment, uses a well defined acoustic channel and is not under an emotional stress. Moreover, the acoustic model as well as language model in the used ASR system can be tuned specifically for the given speaker. On the top of it, one or more human correctors usually correct misrecognized words, add punctuation, perform hyphenation (if needed) and format the recognized text to the final captions. With this setup, the reported accuracies are usually highly over 95 %.

As already said, one of the main objectives was development of a captioning system supporting real-time online operation, generating captions either from the real acoustic track or from the track respoken by the shadow speaker. Since there were no skilled shadow speakers available in the beginning of the project, we have identified several TV shows that we found captionable using the real acoustic track. The set of suitable shows contained weather news, specific discussion shows and meetings of the Chamber of Deputies and the Senate of the Parliament of the Czech Republic. After discussion with representatives of the CTV, we decided to pursue captioning of the meetings of the Chamber of Deputies of the Czech Parliament.

The Chamber of Deputies has 200 members, elected every four years. Although the number of members is quite large, only a small portion of them acts actively during the meetings. Moreover, a large portion of the active speakers stays in the service several electoral terms. These speakers are skilled orators and the rules of procedure enforce that no member may speak unless called upon to do so by the chairman. The audience chamber has a professional, high quality audio capture system and the acoustic channel is stable and the quality is sufficient for ASR.

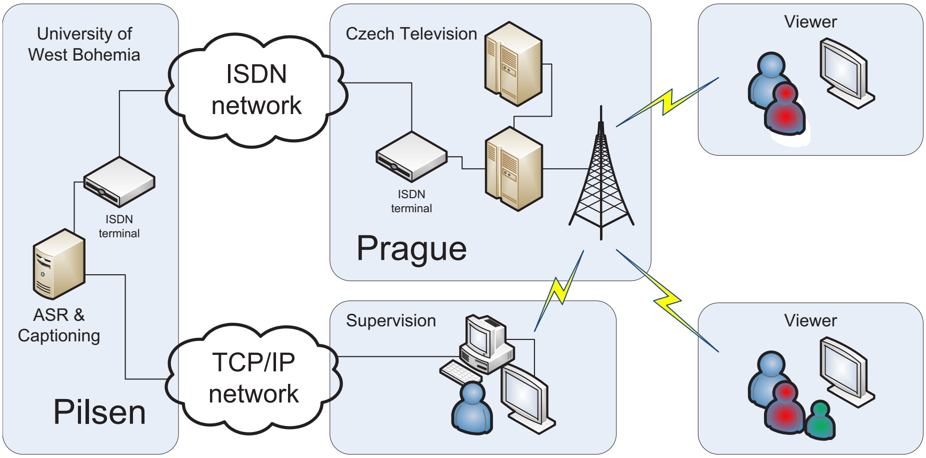


Fig. 1. Geographic scheme of the pilot captioning system

2 Captioning System Architecture

Because of the strict security policies at the premises of Czech Television, the system was designed as highly distributed, see Fig. 1. The interconnection between CTV and UWB is done by a point-to-point connection over the ISDN network. However, instead of using the ISDN voice services, the ISDN is used only as data carrier. The two B channels of the ISDN-BRI are bonded, providing bandwidth of 128 kbit/s.

To provide transparent interconnection over the ISDN network, specialized terminal adapters CDQPrima 210 are used. Outside of the security consideration, this design decision helped us to isolate ourselves from the network issues and audio streaming issues. The resulting formatted captions are sent over a serial line back to the codec and then over the ISDN network using the "ancillary data" feature of the codecs. Then they are broadcasted as EIA-608 (line 21) captions available using teletext page 888.

The ISDN audio codecs support quite a large variety of audio compression standards. We evaluated several supported compression standards: MPEG-1 Audio Layer II, MPEG-1 Audio Layer III and G.722. We evaluated the suitability of each of this codecs by transcoding the training audio data by a software implementation of the specific codec, training the acoustic models and performing a recognition tests on the heldout data. Because of the recognizer accuracy, we have originally chosen the MPEG-1 Audio Layer III standard however because of technical reasons not tied to, the 128 kbit/s, 48 kHz MPEG-1 Audio Layer II (MP2) codec is used in the current pilot system. The recognizer accuracy drop is not fatal and the MP2 standard offers lower algorithmic delay.

The time delay from word utterance to receiving the data by broadcast receivers is about 2–4 seconds. For the detailed analysis, see Fig. 2. It is clear, that the most prominent factor of the delay is defined by the caption formatting and postprocessing subsystem. However, this delay is dictated by the length of subtitle itself, since the caption is formed from the recognized text spanning the mentioned time interval.

3 Speech Recognition

During the preparatory phase of the project, we gathered about 200 hours of parliamentary sessions recordings. The training data were obtained by means of recording

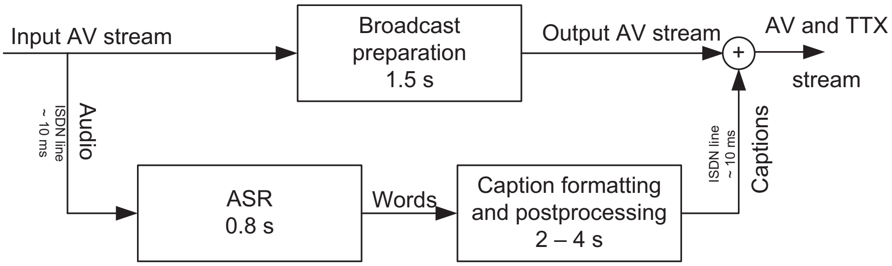


Fig. 2. Timing scheme of the pilot captioning system

the broadcasting. We use three-state HMMs and 8 mixtures of multivariate Gaussians for each state. The total number of approx. 50k Gaussians is used for the speaker-independent model. In addition, discriminative training techniques were used (see [5]). The acoustic model were trained on 44.1 kHz audio stream using PLP parametrization with 19 PLP filters and 12 PLP cepstral coefficients, augmented by delta and delta-delta parameters. To model the transport channel the source data were transcoded (encoded and decoded again) using a software implementation of the 128 kbps MPEG-1 Audio Layer II compression codec.

For language modeling we used the stenographic transcripts that are made public by law. To allow subtitling of arbitrary (including future) electoral period, five classes for representative names in all grammatical cases were created. See [6] and [7] for details. The vocabulary size is approx. 200k words. For the fast online recognition, we use a class-based bigram language model with modified Knesser-Ney discounting trained by SRI Language Modeling Toolkit. For a more accurate confidence measure of recognized words, the class-based trigram language model is used.

Before each captioning session, the language model is adapted using public materials from world-wide web – we use the related texts to integrate the new words and the related n-grams.

The speech recognition accuracy is variable, depending on the type of procedure, orating skills, states of mind and tempers of the speakers and of course the topic. In overall, it is about 85 %–88 %. When the flow of the meeting is highly directed by the rules of procedure (for example voting), the recognition accuracy is over 90 %. The majority of the recognition errors is caused by missing, redundant or misplaced prepositions – according to our experience this kind of error does not hamper the legibility of the subtitles. Moreover, some of these errors can be corrected by postprocessing.

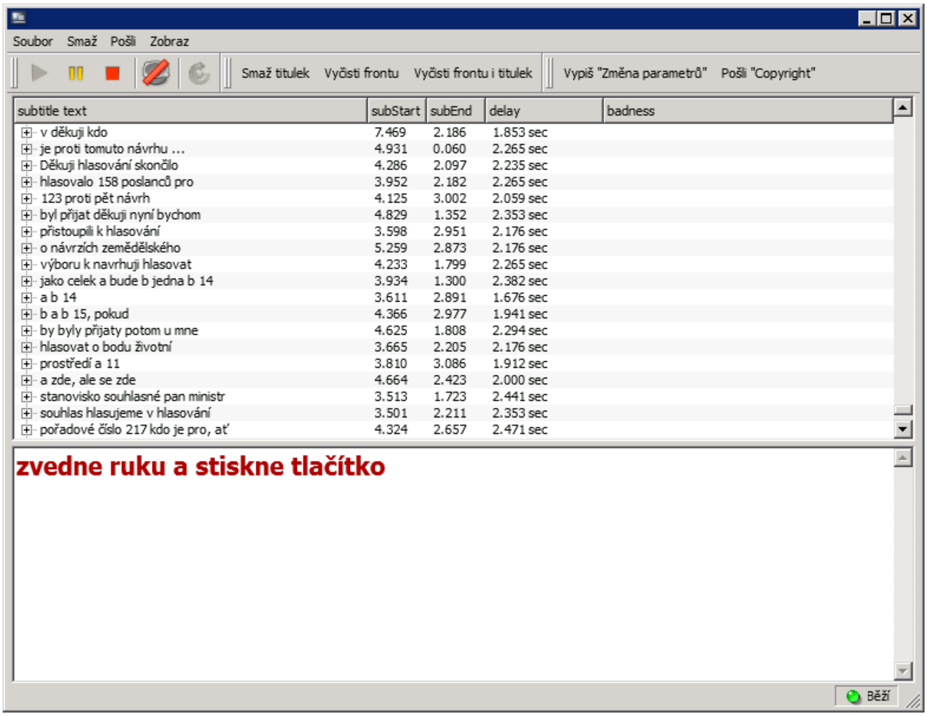
3.1 Automatic Caption Generation

We developed two versions of caption formatting and postprocessing (CFP) subsystem. The first version was developed as a proof-of-concept. It was single user application only, the user had to use the RDP protocol client to log on the captioning server and the possibilities of user assisted caption editation or corrections were limited. The listening-in feature was also accomplished by the RDP client-server architecture.

The development of the second version of the CFP subsystem started after the pilot system evaluation (see below). We have designed a fully distributed client-server application that takes the comments and suggestions of the interviewees into account as well.

The server is the computer where the ASR and the CFP run. It supports automated operation without intervention of any kind. The client is any computer that runs the client software. The client software supports the control of the server (starting, stopping, pausing of the recognition) and editation of the formatted captions.

The number of client users is unlimited. The communication protocol runs over TCP/IP and is secured via SSL. The clients authenticate themselves using an encrypted client SSL certificates issued by the internal certification authority. This is to ensure secure operation independent on the location of the client.



subtitle text	subStart	subEnd	delay	badness
v děkuji kdo	7.469	2.186	1.853 sec	
je proti tomuto návrhu ...	4.931	0.060	2.265 sec	
Děkuji hlasování skončilo	4.286	2.097	2.235 sec	
hlasovalo 158 poslanců pro	3.952	2.182	2.265 sec	
123 proti pět návrh	4.125	3.002	2.059 sec	
byl přijat děkuji nyní bychom	4.829	1.352	2.353 sec	
přistoupili k hlasování	3.598	2.951	2.176 sec	
o návrzích zemědělského	5.259	2.873	2.176 sec	
výboru k navrhuji hlasovat	4.233	1.799	2.265 sec	
jako celek a bude b jedna b 14	3.934	1.300	2.382 sec	
a b 14	3.611	2.891	1.676 sec	
b a b 15, pokud	4.366	2.977	1.941 sec	
by byly přijaty potom u mne	4.625	1.808	2.294 sec	
hlasovat o bodu životní	3.665	2.205	2.176 sec	
prostředí a 11	3.810	3.086	1.912 sec	
a zde, ale se zde	4.664	2.423	2.000 sec	
stanovisko souhlasné pan ministr	3.513	1.723	2.441 sec	
souhlas hlasujeme v hlasování	3.501	2.211	2.353 sec	
pořadové číslo 217 kdo je pro, ať	4.324	2.657	2.471 sec	

zvedne ruku a stiskne tlačítko

Fig. 3. Main screen of the subtitling software

The communication protocol is proprietary, message oriented. To support the listening-in feature even over low-bandwidth internet connection, the server compress the original acoustic track using the Speex codec and streams the compressed track instead of the original track.

Caption types. The standard EIA-608([8]) defines three kinds of closed captions:

- Roll-up – the words appear one by one until they fill the line, then the line is rolled up (possibly erasing the line on very top) and the new line is started to be filled. This is the recommended format of captions for live events, where the captions are generated on-the-fly.
- Pop-on, block – the captions are pre-formatted to span one or several lines and appear instantaneously, erasing the previous captions block. This kind of captions is used prevailingly nowadays.
- Paint-on – the captions appear on the screen by letter-by-letter fashion, forming a stationary block in the end, just like the pop-on captions. Almost unused nowadays.

Because of CTV policies, the captions formatting subsystem we developed produces the pop-on captions. By custom, the captions are displayed as block spanning one or two lines, depending on speaker speech cadence.

To produce eye-pleasing captions we developed a special formatting algorithm based on dynamic programming. The algorithm ranks all the possible variants of captions generated from the given block of recognizer results according to their amenity and chooses the best. If even the best caption scores lover than a certain threshold, the caption is not produced that time. There are several factors that are taken into account when evaluating the amenity:

- delay from the previous caption – too short or too long delays are penalized
- ratio of line filling – shorter lines are penalized
- ratio of line lengths – evaluated only in case where the proposed caption spans two lines, to obtain lines of approximately the same size
- line breaking rules – caption line is not allowed to end with preposition
- internal CTV policies
- ad-hoc rules

Moreover, the caption variant formatting process performs rule-based editation of the resulting caption. During this phase, punctuation is filled in and some kinds of recognizer errors and speaker tics are corrected. The corrected recognizer errors include stray prepositions, multiple prepositions, the corrected speaker tics consisting of word repetitions, hesitation sounds, etc.

In the pilot version of the captioning system, the editation rules were produced by an expert, however, we are currently working on an automatic and a semi-automatic inference of the editation rules.

4 Evaluation of the System among the Targeted Audience

Several months after the pilot captioning system was deployed, an evaluation of the system was arranged. The questionnaires consisted of 7 questions formed into a "single choice from multiple choices" form available online on the web.

1. Do you think the project shall continue?
2. If you have seen the captioned broadcast, do you think the captions were understandable?
3. Do you see the errors in punctuation as a big problem?
4. Did the delay hamper the viewing experience?
5. Do you think that even a partial correction of the biggest errors is beneficial?
6. What kind of captions would you prefer?
7. What kind of speaker change notification would you prefer?

Moreover, the interviewees were asked to append their own comments or questions. We find that the evaluation was necessary to obtain the perspective of the potential end-users.

The findings were as following. The majority of the responders found the subtitles comprehensible and does not find the 2–5 seconds delay to be a problem. The same holds for the punctuation and for the error correction – responders do not feel any of it is vital. Note that this is most probably because of the nature of the programme. Very important were the answers to the last two questions. A majority of the responders prefer the speaker change notification to be accomplished via caption color change and is in favor of the roll-up subtitles. From the additional comments it is clear that the most problematic aspect of the subtitling is combination of the rolling realtime information line that is overlayed over the original program during the broadcast preparation in CTV. The responders found it very uncomfortable, since even without the infoline they have to split their attention between the original video and the subtitles and scrolling of the infoline disturbs the viewers' attention. This can be alleviated by rendering a full-size black box around the subtitles. Lastly, one comment suggested the automatically captioned subtitles should use different pictogram than the programmes with manual captions.

5 Conclusion and Future Prospects

In the paper we presented an overview of a captioning system used by a public service broadcaster in the Czech Republic. The captioning system supports real-time online captioning either from the real acoustic track or the acoustic track produced by the shadow speaker. The system is fully functional; however we plan to enhance it in several ways. Besides improvements of the language model adaptation process and automatic switching between gender-specific and speaker-dependent acoustic models we aim to simplify and possibly eliminate the necessity of human assisted caption correction.

This includes generation of automatic correction rules from large text base, possibly using methods of automatic translation. The semantic punctuation can be supplemented by means enhancing the automatic correction subsystem to use prosodic and acoustic features.

References

1. Evans, M.J.: Speech Recognition in Assisted and Live Subtitling for Television. White Paper WHP065, BBC Research & Development (2003)
2. Homma, S., Kobayashi, A., Oku, T., Sato, S., Imai, T., Takagi, T.: New Real-Time Closed-Captioning System for Japanese Broadcast News Programs. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 651–654. Springer, Heidelberg (2008)
3. Saraclar, M., Riley, M., Bocchieri, E., Goffin, V.: Towards Automatic Closed Captioning: Low Latency Real Time Broadcast News Transcription. In: Seventh International Conference on Spoken Language Processing (2002)
4. Boulianne, G., Beaumont, J., Boisvert, M., Brousseau, J., Cardinal, P., Chapdelaine, C., Comeau, M., Ouellet, P., Osterrath, F.: Computer-Assisted Closed-Captioning of Live TV Broadcasts in French. In: Ninth International Conference on Spoken Language Processing (2006)
5. Povey, D.: Discriminative Training for Large Vocabulary Speech Recognition. Ph.D. thesis, Cambridge University, Engineering Department (2003)
6. Pražák, A., Müller, L., Psutka, J. V., Psutka, J.: LIVE TV SUBTITLING – Fast 2-pass LVCSR System for Online Subtitling. In: SIGMAP 2007, International Conference on Signal Processing and Multimedia Applications (2007)
7. Pražák, A., Psutka, J., Hoidekr, J., Kanis, J., Müller, L., Psutka, J.: Automatic Online Subtitling of the Czech Parliament Meetings. In: newblock Lecture Notes in Artificial Intelligence. LNCS (LNAI), pp. 501–508. Springer, Heidelberg (2006)
8. EIA-608-B: Recommended Practice for Line 21 Data Service. Technical Report EIA/ANSI 608-B, Electronic Industries Alliance (1994)

Adaptation of a Feedforward Artificial Neural Network Using a Linear Transform*

Jan Trmal, Jan Zelinka, and Luděk Müller

Department of Cybernetics, University of West Bohemia
306 14, Plzeň, Czech Republic
{jtrmal,zelinka,muller}@kky.zcu.cz

Abstract. In this paper we present a novel method for adaptation of a multi-layer perceptron neural network (MLP ANN). Nowadays, the adaptation of the ANN is usually done as an incremental retraining either of a subset or the complete set of the ANN parameters. However, since sometimes the amount of the adaptation data is quite small, there is a fundamental drawback of such approach – during retraining, the network parameters can be easily overfitted to the new data. There certainly are techniques that can help overcome this problem (early-stopping, cross-validation), however application of such techniques leads to more complex and possibly more data hungry training procedure.

The proposed method approaches the problem from a different perspective. We use the fact that in many cases we have an additional knowledge about the problem. Such additional knowledge can be used to limit the dimensionality of the adaptation problem.

We applied the proposed method on speaker adaptation of a phoneme recognizer based on TRAPS (Temporal Patterns) parameters. We exploited the fact that the employed TRAPS parameters are constructed using log-outputs of mel-filter bank and by virtue of reformulating the first layer weight matrix adaptation problem as a mel-filter bank output adaptation problem, we were able to significantly limit the number of free variables. Adaptation using the proposed method resulted in a substantial improvement of phoneme recognizer accuracy.

1 Introduction

Nowadays, the MLP ANN are increasingly used in the speech recognition field. Their uses include applications in speech recognition tasks, discriminative features production, language modeling, etc. The main characteristic is that the networks have very large number of parameters (hundreds of thousands or millions).

Given the size of the network, the training and retraining phases are computationally demanding and the amount of data needed during these phases is significant. This is not necessarily a problem during the training phase. However, there are situations, where

* This research was supported by the Ministry of Education of the Czech Republic, project No. MŠMT LC536 and by the Grant Agency of the Czech Republic, project No. GAČR 102/08/0707 and by the University of West Bohemia, project No. SGS-2010-054. The access to the METACentrum computing facilities provided under the research intent MSM6383917201 is appreciated.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 423–430, 2010.

the possibility of fast retraining on fresh data is beneficial. Moreover, the amount of the fresh data is usually quite small.

The limited volume of available data is an obstacle that renders the speaker adaptation and speaker adaptive training paradigms common in HMM ASR field very difficult to implement. The main issue is in the disbalance between the limited amount of available adaptation data (several hundred feature vectors) and large number of free variables. The common training algorithms under these conditions tend to heavily overtrain the network. This problem can be circumvented to some extend by employing cross-validation, early stopping and similar approaches, but it complicates the training process and increases the demands on amount of data.

In this paper we present a general method that enables adaptation of the weight matrix of the first layer of the ANN even if only a small amount of data is available. This is made possible by limiting the number of adaptation parameters. In many cases we can use an additional knowledge about the structure of feature vector, nature of the task, etc. to enforce an inner structure of the adaptation matrix and thus limit the number of free variables. Doing so enables substantial performance improvements through adaptation even on small data sets.

2 Multi-layer Perceptron Artificial Neural Network

Any forward operation of a L-layer MLP ANN can be described as follows

$$\boldsymbol{a}_{\mathbf{0}}(k) = \boldsymbol{x}(k)\boldsymbol{W}_0 \tag{1}$$

$$\left.\begin{aligned} \boldsymbol{y}_i(k) &= \vec{g}_i(\boldsymbol{a}_{i-1}(k)) \\ \boldsymbol{a}_i(k) &= \boldsymbol{y}_i(k)\boldsymbol{W}_i \end{aligned}\right. \qquad i = 1, \dots, L-1 \tag{2}$$

$$\boldsymbol{z}(k) = \vec{g}_L(\boldsymbol{a}_{L-1}(k)) \tag{3}$$

where the $D_i \times D_{i+1}$ matrices $\boldsymbol{W}_i$, $i = 0, \dots, L-1$, are called weight matrices and the vector functions $\vec{g}_i$, $i = 1, \dots, L-1$, are called transfer functions. The weight matrices are trained to minimize a loss function $\boldsymbol{E}$ that is usually of the following form

$$\boldsymbol{E}(\boldsymbol{Z}, \boldsymbol{T}) = \sum_{k=0}^{K} E(\boldsymbol{z}(k), \boldsymbol{t}(k)) \tag{4}$$

where K is the total number of training examples, the $K \times D_L$ matrix $\boldsymbol{Z}$ represents network outputs and the $K \times D_L$ matrix $\boldsymbol{T}$ represents the target values (teacher data). The pair of k-th rows of the matrices $\boldsymbol{Z}$ and $\boldsymbol{T}$ represents the k-th output $\boldsymbol{z}(k)$ and the target vector $\boldsymbol{t}(k)$.

The most usual choices of function E are E_{MSE} (i.e. mean square error) or cross-entropy E_{XENT}. See [1] for more info.

For the training there is a wide variety of methods to use. The most common one is the backpropagation and its modifications.

3 Adaptation of the Neural Network

The most straightforward approach is just to treat the adaptation data just like plain training data and the original set of weight matrices W_i, $i = 0, \dots, Q$ as a "good initialization". Then adaptation of the old ANN means simply retraining the old ANN on new data.

We propose another approach, inspired by the weight-sharing approach used occasionally to improve generalization of ANN. Let's begin with expressing the new weight matrix $\boldsymbol{W}_0$ as

$$\boldsymbol{W}_0 = \boldsymbol{\Gamma} \boldsymbol{W}'_0 \tag{5}$$

where $\boldsymbol{W}'_0$ represents the old weight matrix and $\boldsymbol{\Gamma}$ is an *adaptation matrix*. The overtraining phenomena during the adaptation phase can be reduced or overcomed by virtue of enforcing an inner structure of the adaptation matrix $\boldsymbol{\Gamma}$, thus limiting the number of free variables that must be determined.

Suppose the D_0 by D_0 adaptation matrix $\boldsymbol{\Gamma}$. Because of its assumed inner structure, we can express the matrix as a function of a S-dimensional vector $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_S]$.

$$\boldsymbol{\Gamma} = \boldsymbol{\Gamma}(\boldsymbol{\gamma}) \tag{6}$$

or, expressed alternatively

$$\boldsymbol{\Gamma} = \begin{pmatrix} \Gamma_{11}(\boldsymbol{\gamma}) & \Gamma_{12}(\boldsymbol{\gamma}) & \dots & \Gamma_{1D_0}(\boldsymbol{\gamma}) \\ \Gamma_{21}(\boldsymbol{\gamma}) & \Gamma_{22}(\boldsymbol{\gamma}) & \dots & \Gamma_{2D_0}(\boldsymbol{\gamma}) \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{D_01}(\boldsymbol{\gamma}) & \Gamma_{D_02}(\boldsymbol{\gamma}) & \dots & \Gamma_{D_0D_0}(\boldsymbol{\gamma}) \end{pmatrix} \tag{7}$$

Therefore, instead of computation of $D_0 \times D_0$ parameters we have to determine S parameters. Note that this is without any loss of generality, since S may be even bigger than $D_0 \times D_0$.

We want to minimize the criterion (4). Attempt to minimize it directly leads to application of either gradient-less optimization method (i.e. Powell's algorithm, Nelder-Mead simplex algorithm), or gradient method where gradient is replaced by its approximation. Both these approaches are usually a last-resort solution, because of their slow convergence and computation demands. Luckily, the expressions of gradient computations are quite easy to obtain.

Applying the matrix derivative chain rule for $\frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{\gamma}}$ we get

$$\frac{\partial E}{\partial \gamma_i} = \mathrm{Tr}\left[\left(\frac{\partial E}{\partial \boldsymbol{\Gamma}} \right)^{\mathrm{T}} \frac{\partial \boldsymbol{\Gamma}}{\partial \gamma_i} \right] \qquad \text{for} \quad i = 1, \dots, S \tag{8}$$

where the computation of the expression $\frac{\partial \boldsymbol{\Gamma}}{\partial \gamma_i}$ is straightforward, since by definition the $\Gamma_{kl}(\boldsymbol{\gamma})$ is known for every element Γ_{kl} of the matrix $\boldsymbol{\Gamma}$.

The expression $\frac{\partial E}{\partial \boldsymbol{\Gamma}}$ can be determined by a similar approach as used for backpropagation. Using the equations (5) and (1) – (3) to compute $\frac{\partial E(k)}{\partial \tilde{\boldsymbol{\Gamma}}}$ and applying the derivation chain rule, we arrive to the following expression

$$\frac{\partial E(k)}{\partial \boldsymbol{\Gamma}} = \frac{\partial E}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{a}_{L-1}} \prod_{i=L-1}^{1} \left(\boldsymbol{W}_i^{\mathrm{T}} \frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{a}_{i-1}} \right) \boldsymbol{W'}_0^{\mathrm{T}} \boldsymbol{x}(k) \tag{9}$$

The derivatives $\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{a}_{i-1}}$ and $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{a_{L-1}}}$ are $D_i \times D_i$ ($D_L \times D_L$ respectively) matrices $\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{a}_{i-1}} = (\sigma_{kl})$, where the element σ_{kl} at coordinates (k, l) is given by

$$\sigma_{kl} = y_i \delta_{kl} - y_k y_l \tag{10}$$

in case when g_i is a softmax transfer function and

$$\sigma_{kl} = \delta_{kl} y_k (1 - y_l) \tag{11}$$

in case when g_i is a sigmoidal transfer function.

For the error function expression $\frac{\partial E}{\partial z}$ the following expressions holds

$$\frac{\partial E_{\mathrm{XENT}}}{\partial z_{ij}} = -\delta_{ij} \frac{t_i}{z_j} \tag{12}$$

$$\frac{\partial E_{\mathrm{MSE}}}{\partial z_{ij}} = t_i - z_j \tag{13}$$

4 Phoneme Recognition and TRAPS Parametrization

The used TRAPS feature vectors are constructed from the log-output of mel-filter bank. The process of the construction is described in detail in [2], assume here for the sake of simplicity the following approach. The process is depicted on Figure 1.

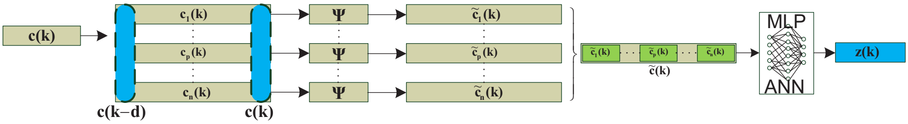


Fig. 1. A scheme of TRAPS phoneme recognizer

Assume for the given k-th frame of speech, the mean-normalized vector of log-outputs from the mel-filter bank is $\boldsymbol{c}(k) = [c_1(k), c_2(k), \ldots, c_N(k)]$, where N is the number of frequency bins.

The vector of D consecutive outputs of the p-th filter bank $\boldsymbol{c}_p(k) = [c_p(k - D + 1), \ldots, c_p(k)]$, $p = 1, \ldots, N$ is then decorrelated by a $D \times N_\Psi$, $N_\Psi \leq D$ matrix Ψ

$$\tilde{\boldsymbol{c}}_p(k) = \boldsymbol{c}_p(k) \cdot \Psi \qquad p = 1, \ldots, N \tag{14}$$

Usually, the Ψ matrix is a discrete cosine transform matrix. The vectors $\tilde{\boldsymbol{c}}_p(k)$, $p = 1, \ldots, N$, are merged together, yielding the TRAPS vector $\tilde{\boldsymbol{c}}(k)$ of size M, $M = N_\Psi N$, $\tilde{\boldsymbol{c}}(k) = [\tilde{\boldsymbol{c}}_1(k), \ldots, \tilde{\boldsymbol{c}}_N(k)]$. The vector $\tilde{\boldsymbol{c}}(k)$ is then used as the input $\boldsymbol{x}(k)$ in expression 1. Therefore, obviously, $D_0 = M$.

The TRAPS feature vectors are usually quite long, since they span several hundreds of milliseconds of the original acoustic track. The features are fed into the ANN trained to produce phoneme posteriori probabilities. The ANN has two-layer design, with a sigmoid transfer function in the hidden layer and a softmax transfer function in the output layer.

5 Speaker Adaptation of ANN

There is a significant variability among speakers. Among other reasons, the cause of variability is a different length of the vocal tract. The different length of the vocal tract manifests itself in shift of the formant center frequencies. There is a variety of methods that deal with this problem.

One of the simplest methods is called VTLN (Vocal Tract Length Normalization). VTLN applied on mel filters compensate the pitch shift by warping of the frequency spectrum. However, it has been shown (see [3] and [4]) that VTLN can be represented as a linear transform of the original, unnormalized coefficients. Therefore, the linear transform can be used for speaker normalization, even without linking it directly to VTLN. The normalized ("adapted") frequency bin $c_i(k)$ (using the notation from previous section) is obtained from the old ("unadapted") frequency bin $c'_i(k)$

$$c_i(k) = \sum_{j=1}^{N} \gamma_{ij} c'_j(k) \qquad i = 1, \ldots, N \tag{15}$$

and the coefficients γ_{ij} must be determined during the speaker normalization phase of the training process. As can be seen, the coefficients γ_{ij} form a $N \times N$ matrix $\boldsymbol{\gamma}$

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{N1} & \gamma_{N2} & \cdots & \gamma_{NN} \end{pmatrix} \tag{16}$$

Depending on the amount of adaptation data, we can limit the number of coefficients, starting from full matrix, going through various banded matrices and ending with a diagonal matrix.

From the description of simplified TRAPS construction process we can infer the structure of adaptation matrix $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}(\boldsymbol{\gamma})$. The resulting matrix $\boldsymbol{\Gamma}$ will have a remarkably similar structure

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\gamma}_{11} & \boldsymbol{\gamma}_{12} & \cdots & \boldsymbol{\gamma}_{1N_\Psi} \\ \boldsymbol{\gamma}_{21} & \boldsymbol{\gamma}_{22} & \cdots & \boldsymbol{\gamma}_{2N_\Psi} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\gamma}_{N_\Psi 1} & \boldsymbol{\gamma}_{N_\Psi 2} & \cdots & \boldsymbol{\gamma}_{N_\Psi N_\Psi} \end{pmatrix} \tag{17}$$

however the matrix element $\boldsymbol{\gamma}_{ik}$ represents a $N \times N$ diagonal matrix

$$\boldsymbol{\gamma}_{ik} = \gamma_{ik}\mathbf{I} \tag{18}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix.

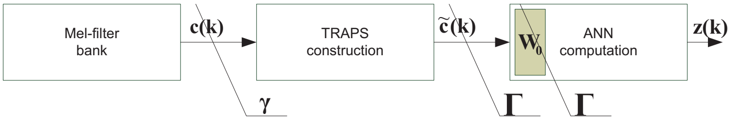


Fig. 2. A scheme of possible ANN adaptation

The transformation of the vector $\boldsymbol{c}(k)$ by matrix $\boldsymbol{\gamma}$ is therefore equivalent to the transformation of the TRAPS vector $\tilde{\boldsymbol{c}}(k)$ by a matrix $\boldsymbol{\Gamma(\gamma)}$. Different approaches may be suitable in different cases (see Fig. 2)

- Mel-filter bank fixed and ANN weights fixed – use matrix Γ to transform the TRAPS features $\tilde{\boldsymbol{c}}(k)$
- Mel-filter bank fixed and TRAPS fixed – use matrix Γ to transform $\boldsymbol{W}_0$
- TRAPS fixed and ANN weights fixed – use matrix γ to transform $\boldsymbol{c}(k)$.

6 Experiments

The experiments were done on the telephone speech corpus SpeechDat-East. The corpus contains utterances from about 1,000 speakers. For training of the ANN, we used only the phonetically balanced sentences (ID S0–S9, X0–1). Note that some speakers are not represented by a complete set of the 12 phonetically balanced sentences. The ANN topology was $330 \times 1,500 \times 111$. The dimension of output layer reflects the phonetic alphabet consisted of 37 phonemes and each phoneme was modeled as a three state unit.

Table 1. Recognizer accuracy (Acc), evaluated including non-speech events, the baseline value is 75.63, R denotes different choices of the regularization constant

	$R = 0$	$R = 10$	$R = 100$	$R = 500$	$R = 1,000$
band_all	75.30	75.30	75.54	75.92	75.87
band_nosil	**76.54**	76.45	76.30	76.11	75.63
diag_all	74.92	75.06	74.92	76.02	75.73
diag_nosil	75.87	76.07	75.92	75.49	75.49
diag_vocal	75.87	75.87	75.83	76.26	76.07

Since the speaker independent network was trained on the complete training part of the corpus, we decided to split the testing set. This was to ensure validity of our experiments. From the testing data, we selected only 97 speakers that were represented by all 12 sentences. For each speaker, we divided randomly the appropriate set to a set of 10 utterances and a set of 2 utterances. The sets of 10 sentences were used as the adaptation data and the sets of 2 sentences as the test data.

The audio recordings are about 6–8 seconds long; however the utterances themselves are just about 2–5 seconds long. Taking only the actual speech into account, there was approximately 40 seconds of speech data available to adapt the neural network on. We experimented with different variants of adaptation matrix structure and with different choices of which phoneme classes the adaptation should be performed on.

7 Regularization

We studied the influence of regularization as well. To compute the regularization cost we used the following expression

$$E_r = |\boldsymbol{\gamma} - \boldsymbol{I}| \tag{19}$$

Since the optimal ratio between the training error and the regularization is not known, the E_r is combined with the expression (5) in the following way

$$\boldsymbol{E}(\boldsymbol{Z}, \boldsymbol{T}, \gamma) = \sum_{k=0}^{K} E(\boldsymbol{z}(k), \boldsymbol{t}(k)) + R \cdot E_r(\boldsymbol{\gamma}) \tag{20}$$

where the constant R is called a *regularization constant* and is usually determined by choosing its value from a predefined set of weights.

7.1 Training Process

Because we wanted to test the potential of the introduced approach, we used supervised adaptation of the network parameters. As the teacher data, we used the phone forced alignment of the reference transcript by an HMM alignment tool. After every update of weights, we performed another alignment run. We limited the number of update-realignment cycles to 4.

8 Results

The experiment results are shown in tables Table 2 and Table 1. The names of columns represent the individual combinations of the shape of the matrix $\boldsymbol{\gamma}$ ("diag" represents diagonal matrix, "band" represents tridiagonal matrix) and class of phonemes the adaptation has been performed on ("all" represent adaptation on all phoneme classes, "nosil" represents adaptation on all phonemes except silence and non-speech event classes, "vocal" represents adaptation only on voiced phonemes).

Since the neural network was done on the original TRAPS, we determined the matrix $\boldsymbol{\Gamma}(\boldsymbol{\gamma})$ first. The TRAPS are constructed from outputs of 15 mel-filters. This means that instead of adapting of $330 \times 1500 = 500k$ free parameters, we used only 15 free parameters with the diagonal setup, 43 parameters with the triagonal setup or 225 parameters for the full matrix $\boldsymbol{\gamma}$ setup. For the best combination of regularization constant and adaptation setup, we performed the Wilcoxon signed rank test under the null hypothesis that median of the difference between the unadapted and adapted networks is zero. The p-value was $p = 0.003$, which is sufficient to reject the null hypothesis at the level $\alpha = 0.003$.

Table 2. Recognizer accuracy (Acc), evaluated excluding non-speech events, the baseline value is 73.07, R denotes different choices of the regularization constant

	$R = 0$	$R = 10$	$R = 100$	$R = 500$	$R = 1000$
band_all	73.20	73.11	73.32	73.66	73.45
band_nosil	**74.04**	73.78	73.87	73.49	73.32
diag_all	72.73	72.94	72.90	73.62	73.45
diag_nosil	73.32	73.49	73.24	72.99	72.94
diag_vocal	73.41	73.28	73.49	73.83	73.57

9 Conclusion

In this paper we devised a novel approach to a MLP ANN adaptation. We applied the presented method on speaker-based adaptation of a phoneme recognizer based on MLP ANN. Adaptation of this kind of networks on small amount of data is generally a difficult task because of quite large number of network parameters. Application of the method lead to a significant reduction of the number of free variables, thus alleviating the overtraining problem. On approximately 40 seconds of speaker data we achieved absolute improvement of approximately 1% (4% relative reduction of phone error rate).

References

1. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford (2005) ISBN 0-19-853864-2
2. Schwarz, P., Matějka, P., Černocký, J.: Towards lower error rates in phoneme recognition. LNCS, vol. 2004. Springer, Heidelberg (2004)
3. Pitz, M., Molau, S., Schlüter, R., Ney, H.: Vocal tract normalization equals linear transformation in cepstral space. In: Proceedings of EuroSpeech 2001, pp. 2653–2656 (2001)
4. Pitz, M., Ney, H.: Vocal tract normalization as linear transformation of MFCC. In: Proceeding of EuroSpeech 2003 (2003)

Gender-Dependent Acoustic Models Fusion Developed for Automatic Subtitling of Parliament Meetings Broadcasted by the Czech TV

Jan Vaněk and Josef V. Psutka

Department of Cybernetics, West Bohemia University, Pilsen, Czech Republic
{vanekyj,psutka_j}@kky.zcu.cz
http://www.kky.zcu.cz

Abstract. Gender-dependent (male/female) acoustic models are more acoustically homogeneous and therefore give better recognition performance than single gender-independent model. This paper deals with a problem how to use these gender-based acoustic models in a real-time LVCSR (Large Vocabulary Continuous Speech Recognition) system that is for more than one year used by the Czech TV for automatic subtitling of Parliament meetings that are broadcasted on the channel ČT24. Frequent changes of speakers and the direct connection of the LVCSR system to the TV audio stream require switching/fusion of models automatically and as soon as possible. The paper presents various techniques based on using the output probabilities for quick selection of a better model or their combinations. The best proposed method achieved over 11% relative WER reduction in comparision with the GI model.

1 Introduction

In recent years, there appeared some projects for hearing-impaired people to help them to access to the information contained in acoustic signal especially of mass media. One of those projects is automatic subtitling of live broadcasted teleview. Recently, we introduced the system for automatic subtitling of Parliament meetings that are broadcasted by the Czech Television (ČT). This system is now used for more than one year by the ČT on the channel ČT24 (see details in [1]).

Frequent changes of speakers and the direct connection of the LVCSR system to the TV audio stream brings interesting challenges. This paper describes our effort to build and use gender dependent acoustic models. The gender-dependent acoustic modeling is a very efficient way how to increase the accuracy over a gender independent modeling in LVCSR and has been previously considered in the literature [2]. The most typical applications work in two-passes where in the first pass a gender-detection method is used (based on GMMs or on multilayer perceptrons-MLP) and in the second pass the speech is recognized with the corresponding gender-specific acoustic model [3].

In this paper we proposed a new combination methods for fusion of the acoustic models. These methods were applied on the level of acoustic models output probabilities. In recent years, the huge amount of computations related to acoustic model become negligible due to the increasing computer speed and capacity of computer memory. From

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 431–438, 2010.

that point of view it is possible to compute several acoustic models simultaneously and switch or even combine their output probabilities in real-time applications. We would like to discuss and compare such methods with methods commonly used.

2 Methods

Various techniques for acoustic models switching/fusion were proposed. All techniques were designed for the real-time applications therefore only a small history for actual processed frame is needed. The first two methods are based on pure switching of individual acoustic models. The third method switches output probabilities for each time/state independently through all acoustic models. The other methods are based on evaluated total probability of the actual frame for all acoustic models. Some of the proposed methods use exponential forgetting to smooth probability volatility. The detailed description of the methods follows.

2.1 Frame Arg Max

This method marked as *Frame_argmax* chooses for the actual frame the acoustic model that maximizes given criterion. This criterion can be defined in several ways. The commonly used criterion is output probability from GMM or MLP. Because it was necessary to compute the output probabilities for all states in all acoustic models for other switching and fusion methods anyway, we used the total probability of all states of the acoustic model for the actual frame as our criterion:

$$P(\lambda_k|\boldsymbol{o}_t) = \sum_{i=1}^{I} P_k(s_i|\boldsymbol{o}_t), \tag{1}$$

where the total probability is the sum of the all I states s_i of the acoustical model λ_k and $P_k(s_i|\boldsymbol{o}_t)$ is an output probability of the state s_i of the k-th acoustical model and the feature vector $\boldsymbol{o}_t$ in time t. This criterion has, according to our experiments, similar results as the commonly used criterion based on GMMs. Method *Frame_argmax* chooses for actual frame model with the highest total probability. It means that at first the k_{max} is evaluated as

$$k_{max} = \underset{k\in 1\ldots M}{\arg\max}\, P(\lambda_k|\boldsymbol{o}_t) \tag{2}$$

and thus the new probabilities are

$$\hat{P}(s_i|\boldsymbol{o}_t) = P_{k\_max}(s_i|\boldsymbol{o}_t). \tag{3}$$

where M is number of acoustic models and $\hat{P}(s_i|\boldsymbol{o}_t)$ is the new evaluated state's probability.

2.2 Frame Arg Max with Exponential Forgetting

Because the time behavior of the total probability is volatile, some kind of smoothing should be used. The exponential forgetting is a good choice for the real-time applications. The total probabilities for all models are computed as

$$P_t(\lambda_k) = \alpha P_{t-1}(\lambda_k) + (1-\alpha)P(\lambda_k|\boldsymbol{o}_t), \tag{4}$$

where α parameter was set to 0.95. This value was in the center of optimal region in preliminary experiments. Relation between α value and word error rate were examined and results are shown in Section 5. This method marked as *Frame_argmax_exp* is practically the same as the previous method except for using smoothed total probability $P_t(\lambda_k)$ instead of $P(\lambda_k|\boldsymbol{o}_t)$.

2.3 Independent Maximum

The method marked as *Maximum* puts as the new probability of the state s_i the highest value of all M acoustic models.

$$\hat{P}(s_i|\boldsymbol{o}_t) = \max_{k \in 1 \ldots M} P_k(s_i|\boldsymbol{o}_t). \tag{5}$$

It means that the highest output probabilities are searched for each state s_i though all M acoustic models at every time t.

2.4 Independent Multiplication

The following methods, contrary to the previous ones, are fusion of the output probabilities for states across all available acoustic models. The first method marked as *Multiply* is a simple multiplication of M acoustic models likelihoods for individual state:

$$\hat{P}(s_i|\boldsymbol{o}_t) = \sqrt[M]{\prod_{k=1}^{M} P_k(s_i|\boldsymbol{o}_t)}, \tag{6}$$

where $P_k(s_i|\boldsymbol{o}_t)$ is an output probability of the state s_i of the k-th acoustical model. The M-th root is there used to normalize probability back into original range. This approach is implemented internally as an average in log-likelihood domain.

2.5 Independent Average

The second fusion method marked as *Average* is a simple average of M acoustic models likelihoods for individual state:

$$\hat{P}(s_i|\boldsymbol{o}_t) = \frac{1}{M} \sum_{k=1}^{M} P_k(s_i|\boldsymbol{o}_t). \tag{7}$$

2.6 Weighted Multiplication with Exponential Forgetting

Similar to the the switching methods some kind of smoothing should be used. The last two methods use smoothing via weighted sum or multiplication of all probabilities. The weights in time t are computed as

$$w_t^k = \frac{P_t(\lambda_k)}{\sum_{l=1}^{M} P_t(\lambda_l)}. \tag{8}$$

The method marked as *W_mult_exp* evaluates new probabilities as

$$\hat{P}(s_i|\boldsymbol{o}_t) = \prod_{k=1}^{M} P_k(s_i|\boldsymbol{o}_t)^{w_t^k}. \tag{9}$$

In log-likelihood domain this approach can be implemented more simple as weighted sum of the log-likelihoods with precomputed weights w_t^k.

2.7 Weighted Sum with Exponential Forgetting

The method with exponential forgetting is the last fusion method which is proposed in this paper. It is marked as *W_sum_exp* and it evaluates new probabilities as weighted sum

$$\hat{P}(s_i|\boldsymbol{o}_t) = \sum_{k=1}^{M} w_t^k P_k(s_i|\boldsymbol{o}_t). \tag{10}$$

In summary, the three switching and four fusion methods were proposed. All of them are fitted to real-time processing and do not pose any restriction to the number of acoustic models being used.

There's no need to compute all probabilities of all acoustic models for the first two switching methods. It is necessary to compute only one model in actual time if we have some estimate of total probability of individual models. This estimate can be done via much smaller GMM or with some algorithm using Gaussians pruning of the evaluated HMM model.

For fusion methods all state's probabilities of all models need to be evaluated but pruning or other fast HMM evaluation technique can be used. In addition, in the first stage just single acoustic model can be evaluated and, in the second stage, only small number of relevant states can be evaluated for other acoustic models. By using this scenario the computation burden increases over single-model only slightly.

3 Train Data Description

For acoustic model training a microphone-based high-quality speech corpus was used. This corpus of read-speech consists of the speech of 800 speakers (384 males and 416 females). Each speaker read 170 sentences. The database of text prompts from which the sentences were selected was obtained in an electronic form from the web pages of Czech newspaper publishers [4]. Special consideration was given to the sentences selection, since they provide a representative distribution of the more frequent triphone sequences (reflecting their relative occurrence in natural speech). The corpus was recorded in the office where only the speaker was present. The recording sessions yielded totally about 220 hours of speech.

4 Experimental Setup

4.1 Acoustic Processing

The digitization of an analogue signal was provided at 22.05 kHz sample rate and 16-bit resolution format. The aim of the front-end processor was to convert continuous speech into a sequence of feature vectors. Several tests were performed in order to determine the best parameterization settings of the acoustic data (see [5] for methodology). The best results were achieved using PLP parameterization [6] with 27 filters and 12 PLP cepstral coefficients with both delta and delta-delta sub-features (see [7] for details). Therefore one feature vector contained 36 coefficients. Feature vectors were computed each 10 milliseconds (100 frames per second).

4.2 Acoustic Modeling

The individual basic speech unit in all our experiments was represented by a three-state HMM with a continuous output probability density function assigned to each state. As the number of Czech triphones is too large, phonetic decision trees were used to tie states of Czech triphones. Several experiments were performed to determine the best recognition results according to the number of clustered states and also to the number of mixtures. In all presented experiments, we used 16 mixtures of multivariate Gaussians for each of the 4,922 states. The prime single mixture triphone acoustic model trained by Maximum Likelihood (ML) criterion was made using HTK-Toolkit v.3.4 [8]. Further, three 16 mixtures models were trained from the prime model: gender-independent, male and female. The training procedure has two stages. At first, 16 mixtures models were trained with HTK using ML criterion. At second, final models were obtained via two iterations of MMI-FD discriminative training [9,10].

4.3 Gender Based Splitting

As was presented in [9], the splitting via manual male/female markers need not to be optimal due to several "masculine" female and "feminine" male voices occurring in the training corpora and also because of possible errors in manual annotations. Therefore, an initial splitting (achieved via manual markers) was realigned via automatic clustering algorithm. After this process, two more acoustically homogeneous classes were available for gender-dependent acoustic modeling which was described in previous subsection.

4.4 Test Conditions

The test set consists of 100 utterances from 100 different speakers (64 male and 36 female speakers), which were not included in training data. There were no cross talking or speaker changes during each utterance. This portion of utterances was randomly separated to 10 sets so that each set contains at least one male and one female speaker. This multi-utterances were created in order to simulate real-time speaker changes. All recognition experiments were performed with a bigram back-off language model with Good-Turing discounting. The language model was trained on about 10M tokens of normalized Czech Parliament transcriptions. The SRI Language Modeling Toolkit (SRILM) [11] was used for training. The model contains 186k words and the perplexity of the recognition task was 12,362 and OOV was 2.4% (see [12] and [13] for details).

5 Results

To follow up our last year paper [9], the same three acoustic models were used: gender-independent (GI), male and female. At first, all these models were tested stand alone. At second, all switching and fusion method were evaluated. All the results are in Table 1.

Table 1. The results of recognition experiments

Stand alone models	WER [%]
Gender-independent	16.92
Male	22.08
Female	30.07
Switching or fusion	WER [%]
Multiply	17.50
Average	15.54
Maximum	15.47
Frame_argmax	17.36
Frame_argmax_exp	16.41
W_mult_exp	16.83
W_sum_exp	14.96

From the Table 1 it is clear that *Multiply* and *Frame_argmax* methods gave even higher WER than GI model. On the other hand, some methods gave significantly lower WER than GI. The lowest WER has been obtained via *W_sum_exp* method and its relative WER reduction is 2% absolutely and more than 11% relatively.

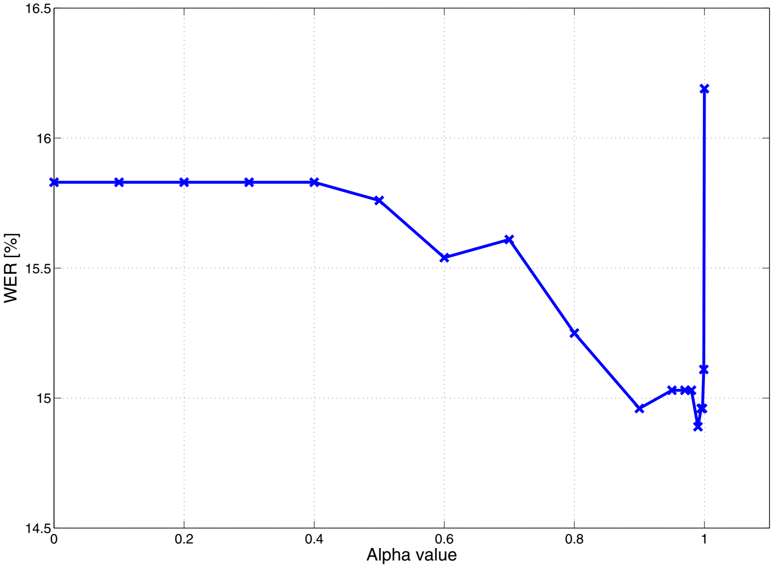


Fig. 1. Relation of α value and WER

Proper setting of α parameter is needed for methods with the exponential forgetting. For all these methods the optimal value range was very similar. The advisable α region is between 0.9 and 0.99. The relation between α value and word error rate is depicted in Figure 1.

6 Conclusion

Various methods of employing gender-dependent acoustic models to the LVCSR system were tested in this paper. The methods had to be designed for real-time automatic subtitling task which is connected to the live TV audio stream. Three switching and four fusion methods were proposed, described and tested. Some of them gave significantly better results than the gender-independent modeling. The lowest WER has been obtained with weighted sum of the HMM state probabilities of all acoustic models (method marked as *W_sum_exp*) and its relative WER reduction is 2% absolutely and more than 11% relatively. All proposed methods are able to combine even higher number of acoustic models than they were tested with.

Acknowledgements

This research was supported by Grant Agency of the Czech Republic, No. 102/08/0707 and by the the Ministry of Education of the Czech Republic, project No. 2C06020.

References

1. Pražák, A., Psutka, J., Hoidekr, J., Kanis, J., Müller, L., Psutka, J.: Automatic Online Subtitling of the Czech Parliament Meetings. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 501–508. Springer, Heidelberg (2006)
2. Olsen, P. A., Dharanipragada, S.: An Efficient Integrated Gender Detection Scheme and Time Mediated Averaging of Gender Dependent Acoustic Models. In: 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003), Geneva, Switzerland (2003)
3. Neto, J., Meinedo, H., Viveiros, M., Cassaca, R., Martins, C., Caseiro, D.: Broadcast News Subtitling System in Portuguese. In: Proceedings of the ICASSP, Las Vegas, USA (2008)
4. Radová, V., Psutka, J.: UWB-S01 Corpus: A Czech Read-Speech Corpus. In: Proceedings of the 6th International Conference on Spoken Language Processing ICSLP 2000, Beijing, China (2000)
5. Psutka, J., Müller, L., Psutka, J. V.: Comparison of MFCC and PLP Parameterization in the Speaker Independent Continuous Speech Recognition Task. In: 7th European Conference on Speech Communication and Technology (EUROSPEECH 2001), Aalborg, Denmark (2001)
6. Hermansky, H.: Perceptual Linear Predictive (PLP) Analysis of Speech. J. Acoustic. Soc. Am. 87 (1990)
7. Psutka, J.: Robust PLP-Based Parameterization for ASR Systems. In: SPECOM 2007 Proceedings. Moscow State Linguistic University, Moscow (2007)
8. Young, S., et al.: The HTK Book (for HTK Version 3.4), Cambridge (2006)
9. Vaněk, J., Psutka, J.V., Zelinka, J., Pražák, A., Psutka, J.: Discriminative Training of Gender-Dependent Acoustic Models. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS (LNAI), vol. 5729, pp. 331–338. Springer, Heidelberg (2009)

10. Vanek J.: Discriminative Training of Acoustic Models. Ph.D. Thesis, West Bohemia University, Department of Cybernetics (2009) (in Czech)
11. Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit. In: International Conference on Spoken Language Processing (ICSLP 2002), Denver, USA (2002)
12. Pražák, A., Ircing, P., Švec, J., et al.: Efficient Combination of N-gram Language Models and Recognition Grammars in Real-Time LVCSR Decoder. In: 9th International Conference on Signal Processing, Beijing, China, pp. 587–591 (2008)
13. Pražák, A., Müller, L., Šmídl, L.: Real-Time Decoder for LVCSR System. In: 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando FL, USA (2004)

Parallel Training of Neural Networks for Speech Recognition

Karel Veselý*, Lukáš Burget, and František Grézl

Speech@FIT, Brno University of Technology
Božetěchova 2, 61266 Brno, Czech Republic
xvesel39@stud.fit.vutbr.cz, {burget,grezl}@fit.vutbr.cz
http://speech.fit.vutbr.cz

Abstract. The feed-forward multi-layer neural networks have significant importance in speech recognition. A new parallel-training tool *TNet* was designed and optimized for multiprocessor computers. The training acceleration rates are reported on a phoneme-state classification task.

Keywords: neural network, phoneme classification, posterior features, backpropagation training, data parallelization.

1 Introduction

Feed-forward multilayer neural networks (NN) have many practical applications [1]. They can be used for classification, pattern recognition, prediction, dimensionality reduction or control. In case of speech recognition, neural networks are typically used as *phoneme classifiers*, where the network input is a vector of features and the output is a vector of phoneme class membership probabilities.

These probabilities have been proven to be very useful for further processing such as direct LVCSR decoding [2], Keyword spotting [3] or Language identification. Sometimes these probabilities are called *posterior features*, because they can be used as features for subsequent systems, or *phoneme-state probabilities*, because they correspond to emission probabilities in standard GMM/HMM approach to speech recognition.

A neural network needs to be trained on data with corresponding frame outputs (in most cases the phoneme labels). The *TNet* tool implements standard *stochastic gradient descent* algorithm with *error backpropagation*. The stochastic variant was chosen because it has better training speed on data-sets with redundant data (eg. with repeating or very similar segments) compared to the batch variant. Moreover stochastic variant gives better possibility of escaping from local minima [4].

Even when using some degree of NN training parallelization, for example as it is implemented in our current training tool *SNet* [5], typically the training time exceeds 24 hours, due to the huge quantities of training data (hundreds hours of speech). The

* This work was partly supported by Grant Agency of Czech Republic project No. 102/08/0707, and by Czech Ministry of Education project No. MSM0021630528, and by Czech Ministry of Interior project No. VD20072010B16.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 439–446, 2010.

long training periods are uncomfortable for practical use. In this paper we describe *TNet* – a new faster implementation of parallel neural network training based on *data parallelization*.

2 Feed-Forward Neural Networks

Feed-forward neural network is an adaptive multivariate transform function with ability to "memorize" patterns by adjusting tunable parameters (neuron weights). It can be seen as a sequence of alternating linear and nonlinear transformations. Due to Kolmorogovs' theorem [1] we believe that the network is able to express any possible function when having enough layers and neurons per layer. In our particular case we are interested in classification NN, the training algorithm will be explained on this class of NN. The *Stochastic gradient descent algorithm* with *error backpropagation* is used for the training, while the weight update is performed per bunch (a block of N frames). In the NN *Sigmoid* nonlinearity is used for hidden layer and *Softmax* nonlinearity is used for the output layer.

For the sake of simplicity, the training algorithm will be explained on the case when the bunch has only one input data-point. The general formula for gradient descent is:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \mu \nabla E \tag{1}$$

which says that the current parameters $\mathbf{w}$ are iteratively moved in the opposite direction of the error function gradient ∇E which is scaled by some learning rate μ. The gradient ∇E of the error function E is a vector of its first derivatives with respect to all the model parameters $\mathbf{w}$:

$$\nabla E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, ..., \frac{\partial E}{\partial w_M} \right]^T \tag{2}$$

To obtain the gradient it is necessary to perform *data propagation*, *objective function evaluation* and *error backpropagation*. The data propagation is done from the first to the last layer:

$$\begin{array}{lll} \text{Linearity} & \text{Sigmoid} & \text{Softmax} \\ \mathbf{y}_n = \mathbf{W}_n \mathbf{x}_n + \mathbf{b}_n & y_{ni} = \frac{1}{1+\exp(-x_{ni})} & y_{ni} = \frac{\exp(x_{ni})}{\sum_j \exp(x_{nj})} , \end{array} \tag{3}$$

where n is the index of linear or nonlinear transform, $\mathbf{y}_n$ is the output vector, $\mathbf{W}_n$ is a neuron-layer weight matrix, $\mathbf{x}_n$ is the input vector and $\mathbf{b}_n$ is the neuron-layer bias vector. Obviously, the output of the previous transformation is input of the next transformation, the input of the first transformation being the input data.

Then the *cross-entropy* error function is evaluated by using the NN output vector $\mathbf{y}_n$ and *desired vector* $\mathbf{d}$. To be able to do the backpropagation, the first derivative of the error function E with respect to NN output vector $\mathbf{y}_n$ must be calculated. In the particular case when we have a coupled *cross-entropy* with *softmax*, the derivative of error function E with respect to softmax input vector $\mathbf{x}_{\text{Softmax}}$ leads to trivial solution which is called *global error*:

$$\begin{array}{ll} \text{Cross-entropy} & \text{Global error} \\ E = -\sum_j d_j \ln(y_j) & \frac{\partial E}{\partial \mathbf{x}_{\text{Softmax}}} = \mathbf{e}_n = \mathbf{y}_n - \mathbf{d} \end{array} \tag{4}$$

Now the error backpropagation can be performed. We start at the last linearity which precedes *Softmax*, proceeding towards the first layer:

$$\begin{array}{ll} \text{Linearity} & \text{Sigmoid} \\ \mathbf{e}_{n-1} = \mathbf{W}_n^T \mathbf{e}_n & \mathbf{e}_{n-1} = \mathbf{y}_n(\mathbf{y}_n - \mathbf{1})\mathbf{e}_n \end{array} \tag{5}$$

Finally the *gradient descent* update formulas are used:

$$\begin{array}{ll} \text{Linearity update} & \text{Bias update} \\ \mathbf{W}_n(t+1) = \mathbf{W}_n(t) - \mu\, \mathbf{e}_n \mathbf{x}_n^T & \mathbf{b}_n(t+1) = \mathbf{b}_n(t) - \mu\, \mathbf{e}_n \end{array} \tag{6}$$

The whole situation of data dependencies can be seen in Fig. 1. The trapezoids represent trainable linear transformations, the circles represent nonlinearities and the dashed lines represent the weight update dependencies.

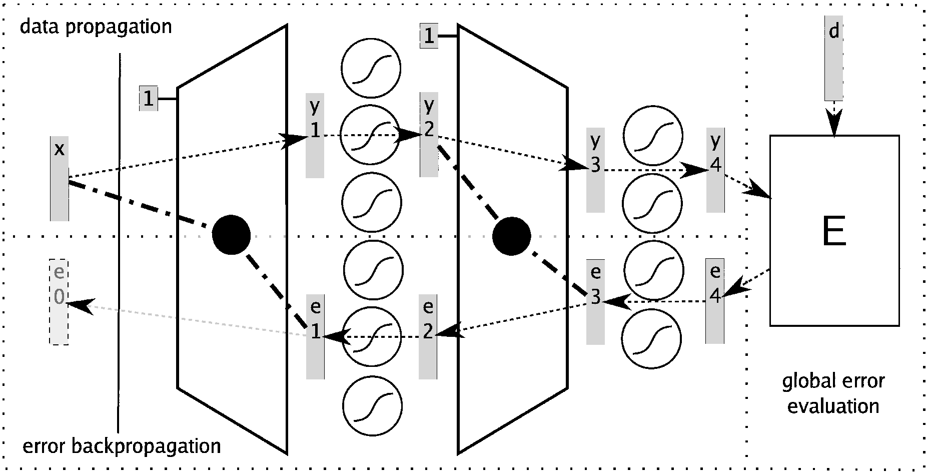


Fig. 1. Data dependencies in the training

The *Stochastic on-line learning* imposes strong data dependencies which makes the parallelization difficult. In our case we are interested in *multi-threading* parallelization which reduces communication overhead due to shared address space of all threads. Two effective approaches to parallel network training have been reported [6]:

Data parallelization The training data is divided into disjoint sets. Each thread has its own network instance and works on its own data-set. Weight synchronization occurs periodically when N frames are processed. The weight difference matrices (eg. error gradients) are gathered, summed up and a new set of weights is generated and distributed.

Node parallelization In this case, there is only one instance of the network. The network layers are divided into disjoint sets of neurons. Each thread has associated its own set. This method imposes higher frequency of synchronization than *data parallelization*. The threads are synchronized by a *barrier* before one can proceed to the next layer.

The problem of *data parallelization* is that the overhead of weight synchronization increases by adding more slave threads. The problem of *node parallelization* is that poor cache performance will slow down the training when layer division sets are too small. A promising strategy could be to combine both approaches and find the optimal operation point.

The question is whether the main performance bottleneck is the performance of the processors or the throughput of the memory controller, in this case it will be useless to use the *data-node parallelization*, and the simpler *data parallelization* would be sufficient.

3 Implementation

Currently the *data parallelization* is implemented in *TNet*. The design of the tool was chosen with respect to both high performance and simple extensibility. The GotoBLAS[1] library is used to accelerate linear algebra operations. The neuron weights are shared for all the threads which improves the processor cache hit-rate.

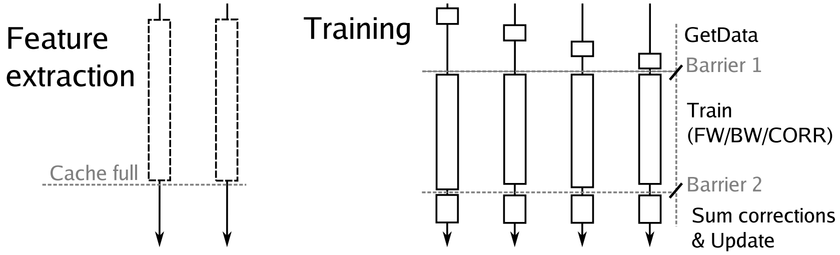


Fig. 2. Thread synchronization (FW – forward pass, BW – backward pass, CORR – weight difference matrices calculation)

As can be seen in Fig. 2, the training process is divided into several training threads and two feature extraction threads. The two feature extraction threads work in background unless the feature cache is full. The training cycle consists of three phases:

1. Data distribution
2. Training (forward pass, objective function evaluation, error backpropagation and collection of weight difference matrices)
3. Merging of weight difference matrices, weight update

The *data distribution* is done in series, because it is simpler and more universal solution. If we do not distribute the data deterministically, we wouldn't be able to repeat the training procedure and get the same network, also the merging of weight update matrices must be deterministic. The *training* and *weight difference matrix merging*

[1] http://www.tacc.utexas.edu/tacc-projects/

is parallel. Two barriers are used to synchronize, one before the training starts and one before the merging starts. Partial summing is used for merging; every thread is responsible for summing several lines of the weight difference matrices. The tool is capable of both *on-line* (Stochastic) and *batch* gradient descent, the only difference is that the weight update is not performed per-cycle but at the end of the epoch.

The *TNet* is compatible with the HTK data formats. It accepts STK[2] transforms for feature extraction, the network is stored in it's own format with possible conversion to the STK format.

After finishing the *TNet* development, the tool will be distributed as Open Source software. Currently the pre-release version can be downloaded at: `http://speech.fit.vutbr.cz/files/software/tnet/TNet.tar.gz`

4 Experimental Results

The baseline parallel NN training implementation is *SNet*. It is an open source tool implemented in 2006 by S. Kontár, it is distributed as part of STK. The training speeds were measured on HP ProLiant DL785 G5 server with 8 quad-core AMD Opteron 8,356 processors (32 cores) and 128GB of RAM.

Database. The training set is a subset of AMI meeting data corpus[3]. The total size of the AMI corpus is 100 hours. A 10h subset was taken as training data-set, the cross-validation was performed on a 1h subset. The corpus is labelled by 45 phonemes.

Parameterisation. The parameters are log Mel filter-bank outputs derived using 25ms window, 10ms shift; 23 filters were used. Such parameters were normalized by Per-Speaker Cepstral Mean-Variance Normalisation and VTLN. Then a 51 frames long context (510ms) was taken for each filter. Each context was separately scaled by Hamming Window and compressed by Discrete Cosine Transform to 26 coefficients. By re-concatenation we get vectors of 598 coefficients. Such network inputs were finally globally Mean-Variance normalized.

Network structure. Two-layer feed-forward NN with one hidden layer was used. The successive layers are fully connected, the dimensionality of input is 598, the hidden layer has 1,000 neurons, the output layer has 135 neurons. The nonlinearities (activation functions) used in the first and the second layer were Sigmoids and Softmax, respectively.

The phonemes are considered context independent with three sub-states, which leads to 135 classes.

Training. Standard backpropagation algorithm with the "newbob" learning-rate scheduling was used: The learning rate is kept fixed as long as the increment in cross-validation accuracy is bigger than a threshold. For the subsequent epochs, the learning rate is being halved till the cross-validation increment is inferior to some stopping threshold.

[2] BUT speech toolkit `http://merlin.fit.vutbr.cz/svn/STK/trunk/src/`

[3] `http://www.amiproject.org/ami-scientific-portal/meeting-corpus`

Table 1. Used slave bunch-sizes

# Slave threads	1	2	3	4	6	8	12	16	20	24
Slave bunch	960	480	320	240	160	120	80	60	48	40
Update bunch	960	960	960	960	960	960	960	960	960	960

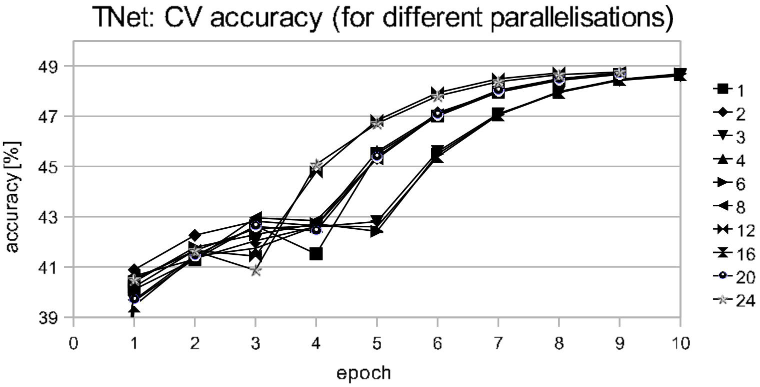


Fig. 3. Cross-validation accuracy convergence, using *TNet* with different parallelizations

The NN weight update was performed per *bunch* (fixed-size block of data-points). The slave bunch-size is different for each parallelization. Since the weight update is done per aggregated bunch-size of all slaves which is equal to original bunch-size of serial training, the two training algorithms are equal. The used bunch-sizes are in Tab. 1. The bunch-size of ∼1,000 frames was proven [5] to be optimal: too small bunch-size causes training slowdown, too big bunch-size causes worse training convergence.

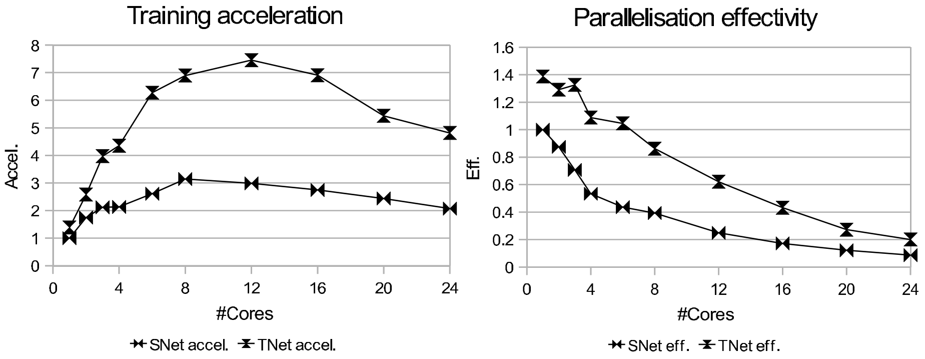


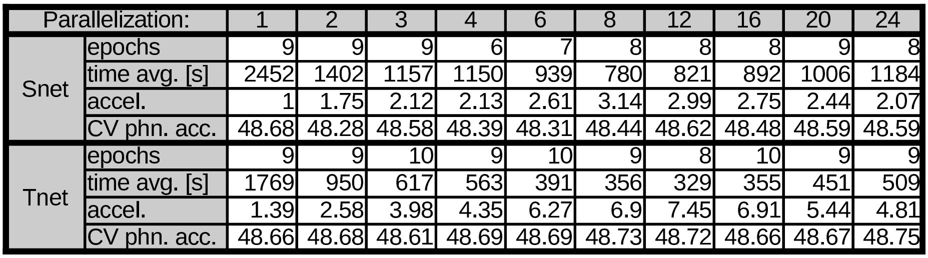
Fig. 4. Acceleration and effectivity of parallelization

Results. Different parallelization orders were tested. In each case, the single-epoch training time was measured at least 8 times. Although the cross-validation accuracies slightly diverged during the training, finally they all converged at ∼48.7% during 8th, 9th or 10th epoch as can be seen in Fig. 3. We believe that this slight divergence is caused by accumulated rounding errors since we are using single float precision for the sake of speed. If the cross-validation accuracy drops, we back-off to previous network and the learning-rate is halved. This is why we see in Fig. 3 curves with depression in center.

The same per-frame phoneme error rate 48.7% was achieved both by *SNet* and *TNet*, as can bee seen in Tab. 2. This proves the equivalence of both implementations.

The obtained parallelization acceleration and effectivity (eg. acceleration / # Cores) is compared in Fig. 4. The recapitulative table with number of epochs, average time-per-epoch, training acceleration and final cross-validation accuracy of per-frame phoneme-state classification is in Tab. 2.

Table 2. Table with summary. "CV phn. acc." is cross-validation accuracy of per-frame phoneme-state classification after last training epoch.



Parallelization:		1	2	3	4	6	8	12	16	20	24
Snet	epochs	9	9	9	6	7	8	8	8	9	8
	time avg. [s]	2452	1402	1157	1150	939	780	821	892	1006	1184
	accel.	1	1.75	2.12	2.13	2.61	3.14	2.99	2.75	2.44	2.07
	CV phn. acc.	48.68	48.28	48.58	48.39	48.31	48.44	48.62	48.48	48.59	48.59
Tnet	epochs	9	9	10	9	10	9	8	10	9	9
	time avg. [s]	1769	950	617	563	391	356	329	355	451	509
	accel.	1.39	2.58	3.98	4.35	6.27	6.9	7.45	6.91	5.44	4.81
	CV phn. acc.	48.66	48.68	48.61	48.69	48.69	48.73	48.72	48.66	48.67	48.75

5 Conclusion

As we can see on the Fig. 4, the *TNet* implementation is more than 2× faster than *SNet* in case of 12 core parallelization. The speed of *SNet* in serial mode was previously proved to be equivalent [5] to speed of *QuickNet*[4] – another very popular NN training tool.

TNet is faster than *SNet* because it uses thread parallelization where all the threads share same address space, while *SNet* is a client-server application which uses TCP-IP connection for weight synchronization.

By adding more than 12 cores to *TNet*, the acceleration decreases. This is possibly caused by limited RAM to CPU bandwidth. Further acceleration of the training is still possible and will be subject of our future work. However, this will require more complicated and less universal design. Another promising way of acceleration is the use of modern GPUs, which can offer massive parallelization.

[4] `http://www.icsi.berkeley.edu/Speech/qn.html`

References

1. Jan, J.: Číslicová filtrace, analýza a restaurace signálů. Vutium (2002) ISBN 80-214-1558-4
2. Karafiát, M., Grézl, F., Schwarz, P., Burget, L., Černocký, J.: Robust heteroscedastic Linear Discriminant Analysis and LCRC Posterior Features in Meeting Data Recognition. LNCS, pp. 275–284. Springer, Heidelberg (2007) ISBN 978-3-540-69267-6
3. Szöke, I., Schwarz, P., Matějka, P., Burget, L., Karafiát, M., Fapšo, M., Černocký, J.: Comparison of Keyword Spotting Approaches for Informal Continuous Speech. In: Proceedings of Interspeech 2005 – Eurospeech (2005) ISSN 1018-4074
4. Bishop, C.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006) ISBN 0-38-731073-8
5. Kontár, S.: Paralelní trénování neuronových sítí pro rozpoznávání řeči. FIT VUT Brno, diplomová práce (2006)
6. Pethick, M., Liddle, M., Werstein, P., Huang, Z.: Parallelization of a Backpropagation Neural Network on a Cluster Computer. In: Parallel and Distributed Computing and Systems. IASTED/ACTA Press (2003)

Design and Implementation of a Bayesian Network Speech Recognizer

Pascal Wiggers, Leon J.M. Rothkrantz, and Rob van de Lisdonk

Man–Machine Interaction Group, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
p.wiggers@tudelft.nl, l.j.m.rothkrantz@tudelft.nl

Abstract. In this paper we describe a speech recognition system implemented with generalized dynamic Bayesian networks (DBNs). We discuss the design of the system and the features of the underlying toolkit we constructed that makes efficient processing of speech and language data with Bayesian networks possible. Features include: sparse representations of probability tables, a fast algorithm for inference with probability tables, lazy evaluation of probability tables, algorithms for calculations with tree-shaped distributions, the ability to change distributions on the fly, and a generalization of DBN model structure.

1 Introduction

A dynamic Bayesian network (DBN) is a probabilistic graphical model that represents the relationships between random variables over time [1]. Among other things, DBNs have been applied to motion and object tracking [2], multi-modal sensor fusion [3], mobile robot navigation [4], prediction of driving behavior [5] and gene expression recognition [6].

It has been argued that dynamic Bayesian networks also provide a good framework for the implementation of speech recognizers and language models [7,8,9] as they can be interpreted as a generalization of both hidden Markov models (HMMs) and n-grams. Unlike these models, Bayesian networks provide a factored state representation, making it easy to add information to the acoustic and language models such as articulatory features [10], visual information on lip movements [11] or contextual information [12].

However, sofar the application of DBNs in speech and language processing has remained limited. This may be due to a lack of toolkits that support the development and implementation of these models. Although numerous toolkits for inference with Bayesian networks exist, most of these were designed with other applications in mind and cannot deal with the requirements posed by tasks like speech recognition and language modeling, such as processing thousands of time slices, working with word variables that can take on 64,000 different values and smoothing of distributions. Also, features offered in software toolkits for the implementation of hidden Markov models such as mixture splitting and pruning are typically not supported.

To fill this gap, we designed and implemented a Bayesian network toolkit called *Gaia* targeted specifically at speech and language processing. In section 4 we will discuss the features of our framework. Section 5 describes how we used the toolkit to implement a

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 447–454, 2010.

speech recognizer, but we start with a definition of Bayesian networks and an overview of related work in the next two sections.

2 Bayesian Networks

Bayesian networks are directed acyclic graphs of which the nodes are random variables and the arcs indicate conditional independence of the variables, i.e. the absence of an arc between two variables signifies that those variables do not directly depend upon each other. Thus a Bayesian network is a factored representation of a joint probability distribution over the set of variables $\mathbf{X} = \left\{X^1, X^2, \ldots, X^n\right\}$ given by:

$$P(\mathbf{X}) = \prod_{X^i} P(X^i | \mathit{Parents}(X^i)). \tag{1}$$

Dynamic Bayesian networks (DBNs) model processes that evolve over time. They are defined by two Bayesian networks: a *prior network* $P(\mathbf{X}_1)$ and a *transition network* $P(\mathbf{X}_t | \mathbf{X}_{t-1})$ that defines how the variables in the set $\mathbf{X}_t$ depend upon each other and on variables in the previous time slice.

3 Related Work

Several toolkits that implement dynamic Bayesian networks exist. We will discuss the prominent ones most similar to our toolkit. dHugin [13] adds temporal reasoning to the popular, commercial Hugin shell. It assumes that DBNs obey the Markov property, i.e. a variable only depends on variables in the current or in the previous time slice. The structure of time slices can vary. An exact junction-tree based inference routine that unrolls the network for k slices at time as well as forward sampling are provided.

The Bayes net toolbox (BNT) [14] is a free, open-source library intended for research purposes. It is implemented in Matlab because of the ease with which it can handle Gaussian random variables. Unfortunately, this choice limits the processing speed and the size of the networks that can be processed. Only first-order Markov processes can be modeled. Several inference algorithms for static Bayesian networks are provided, each of which makes different trade offs between accuracy, generality and speed. Continuous and discrete distributions are supported, as is parameter learning.

The graphical models toolkit (GMTK) [15] is a freely-available toolkit written in C++ that is designed for DBN-based speech recognition. It has many desirable features, such as sparse representations of CPDs, tree-shaped CPDs, continuous observation distributions, switching parents, beam search and generalized EM training. It supports smoothing and Viterbi inference using the online Frontier algorithm. Unfortunately, the length of an input sequence has to be known in advance, making online use impossible.

The Structural Modeling, Inference and Learning Engine (SMILE) [16] is a platform independent library that implements Bayesian networks and influence diagrams. It also supports temporal reasoning.

Even though these toolkits have many interesting features and, like many of their less known competitors, are well-suited for specific domains, none of these systems offers all features necessary for efficient speech and language processing.

4 Design of Gaia

The *Gaia* toolkit consists of a collection of general purpose tools for inference and learning of DBNs. The functionality behind these tools is implemented in a common C++ library that makes it easy to change or reuse parts of the software and to experiment with different algorithms and data structures. The library has a layered structure. At the lowest level it provides general purpose classes, such as input and output routines and memory management. The core of the system is formed by classes that implement efficient mathematical operations on multidimensional probability tables. The top layer is responsible for construction of and inference in DBNs.

Unlike the frameworks discussed above, our framework is developed with the tasks of speech and language processing in mind, but its applicability is not limited to those domains. Specifically, it can deal with very small probabilities – using log probabilities – and a large number of states, while special data representations are provided for sparse and deterministic distributions. The inference algorithms exploit observed values to avoid unnecessary calculations and to keep the size of the resulting table as small as possible. Since probability table manipulation makes up the bulk of the work in probabilistic inference a lot of attention has been dedicated to optimizing this part of the library. For example specialized memory pools are used for fast allocation and deallocation of probability tables.

Inference in Bayesian networks comes down to a sequence of multiplications and marginalizations of likelihood tables. Most inference techniques determine the order of operations based on the structure of the network, without taking the shape of the probability distributions into account. In addition, this is typically done before any evidence – that may introduce additional independence relations – has been observed. As a consequence, the order is not always efficient and might result in unnecessarily large intermediate tables. [17] introduced lazy evaluation in the junction tree algorithm [1] to make better use of evidence and probability table structure. We implemented a similar lazy evaluation approach, but at the level of likelihood tables. This has the advantage that it can be used with any inference algorithm. In this approach multiplication of tables is deferred until it really is necessary, i.e. if a variable in the domain of the table is marginalized out. Rather than a table a set of tables is used. Multiplication is a very fast operation as it simply adds the table to the set. Upon marginalization or summation all tables in the set whose domains contain variables that are marginalized out are selected. Multiplication of these tables and marginalization are done in a single operation to avoid the construction of (large) intermediate tables. Note that if the variables of a particular subtable are never looked at, that table will never really be processed. In Bayesian networks this may occur if certain subparts of the network are not needed in an inference process.

Multinomial distributions are implemented as n-ary decision trees that have these lazy probability tables as their leaves. This way several distributions can share the same subtrees to saves space and time in case of reoccurring substructure, and to implement interpolation and smoothing as well as parameter sharing. In the same way Gaussian components can be shared by several Gaussian mixture models. Complete multinomial and continuous distributions can be shared by several variables in the network.

4.1 Network Structure

The top layer of the library implements a generalization of dynamic Bayesian networks. To increase the flexibility of the system this layer consists of three components: an abstract definition of DBNs that is not bound to any inference algorithm, an interface to the outside world that implements high level inference technique-independent functionality and an inference engine. Different inference engines can be plugged into the system without changing the other structures.

Whereas standard DBNs repeat these subnetwork for every time slice, our networks can have any number of subnetworks, called chapters, that each have a repeating substructure that can span several time slices. The last chapter will repeat indefinitely, all other chapters must have a predefined length.

A chapter can define static variables, that do not have temporal dynamics. It is also possible to define static variables at the network level. Those correspond to the contemporal nodes of [18]. Following [18], we also provide a separate chapter with static variables that is attached at the end of the network when all observations are processed. The parents of a variable can be in any preceding slice or chapter, i.e. k-th order relations are allowed.

Figure 1 shows two chapters and the static end chapter, denoted by rectangles. The first chapter defines the structure of the first time slice. The second chapter is repeated for all following time slices, as indicated by the numbers in the upper right corner of the chapters. The dotted circles are place holders for parents of a node that are defined in other time slices in the expanded network. This network definition, together with the definitions of the (tree-shaped) distributions are provided by the user of the system in XML format. The DBN interface is responsible for expanding (unrolling) this definition into a network and checking it for consistency.

4.2 Inference Engine

The toolkit supports the calculation of the probability of an observation sequence, prediction of the value of a variable based on the history and Viterbi inference. For the latter it can be specified which variables should be marginalized (summed) out and which should be maximized. It is possible to run any of these algorithms in interactive mode to inspect variables during inference. We currently implemented the Frontier algorithm and the Interface algorithm for exact online inference and the Boyen-Koller and factored frontier algorithm for approximate inference [1].

4.3 Learning

For parameter learning the expectation-maximization (EM) algorithm is used. It has been implemented using a forward and a backward pass, enabling the use of any inference engine that implements smoothing. To allow for efficient incremental model improvement, e.g. update parameter weights, it is possible to specify which variables will be updated in a training run and which not. At every time slice intermediate results have to be saved in the forward pass, so the space requirements of the algorithm may quickly get out of hand. Therefore, the Island algorithm [19] has been implemented.

This algorithm saves messages at C island points during the forward pass (including the first and last slice), resulting in $C - 1$ subproblems on which the algorithm is then called recursively. If messages are saved every $\sqrt{T}$ steps the space complexity will be reduced to $O(S \log_C T)$, the cost of this is a increase in time complexity from $O(T)$ to $O(T \log_C T)$.

The learning tool can work in distributed mode. Each processor learns a part of the data. A central thread of the program combines the accumulators of the other programs. Between training runs the defunct mixtures of Gaussian mixture distributions are removed. The number of mixtures can be increased as well.

4.4 Context-Dependent Distributions

A special feature of the toolkit is that the distributions in the network can be changed on the fly. This is especially useful when training with partial information. For example, recordings used for training in speech recognition are typically annotated at the word level. Using a dictionary one can find the sequence of phonemes that are being spoken but it is unknown when a particular word or phoneme starts or ends, therefore this information cannot simply be used as evidence for specific variables (at a specific point in time) during training.

Using the context feature one can change the word distributions in a speech recognizer network such as shown in Figure 1 to match the actual transcription for every recording, i.e. encode the words and the order of the words, while leaving the state and phoneme transitions to be learned.

4.5 Pruning

Inspired by the common practice in hidden Markov models and probabilistic grammars to prune large parts of the search space to achieve real-time performance, we implemented beam pruning that sets all probabilities that fall below some threshold relative to the highest probability in a distribution to zero. For Bayesian networks pruning is seldomly used. Speed ups are achieved using approximate algorithms. The advantage of pruning over such methods is that pruning focuses on the high probability paths through a network given the observations. Stochastic approximate inference techniques such as particle filtering do focus on high probability paths. However, to achieve real-time performance the number of paths through the model that will be sampled will be sparse.

5 A DBN Speech Recognizer

The toolkit was validated by comparing its performance on a large number of networks with that of the BNT toolkit. To further validate the toolkit and show how it can be used for speech recognition we reimplemented a speech recognizer using the toolkit and compared its performance to that of the original HMM-based system. The original system was trained on the Dutch Polyphone dataset [20] using HTK [21]. In a preprocessing phase all audio files were encoded using 12 mel frequency cepstral coefficients (MFCCs) and the total energy in the signal. Those features, together with their first and second order derivatives, were calculated every 10 ms using a window of 25 ms. For this test we used 43 simple 3 state left-to-right HMMs that represented the phonemes of Dutch and models for silence and mouth noises. A 39-dimensional Gaussian distribution over the input is attached to every state.

As, for now, our focus was on the correctness of the algorithms and not on the performance of the recognizer as such, we used a 150 word vocabulary and a wordloop language model, i.e. all words are considered equally likely and every utterance can consist of a sequence of words. Our test set consisted of 181 utterances from the Polyphone set that were not used for training.

The structure of the DBN speech recognizer is shown in Figure 1. In this model the W nodes represent words, i.e. the values of the W variable are the 150 words in our vocabulary. The P nodes represent phonemes. We used the same 43 phonemes as in the baseline system. N^w is a counter variable that keeps track of the phoneme position within a word. Therefore $P(P|W, N^w)$ is a deterministic distribution that encodes that pronunciation dictionary. As before every phoneme has three substate. This is represented by the S variable. Only transitions from a state to itself or to its direct predecessor are non-zero. The O variable represents the state dependent Gaussian distributions over the input. E^p is binary variable that signals that the end of a phoneme is reached. This is only possible if the last state of a phoneme is reached. In the same way E^w indicates the end of a word. $P(E^w|W, N^w, E^p)$ is deterministic, the end of a

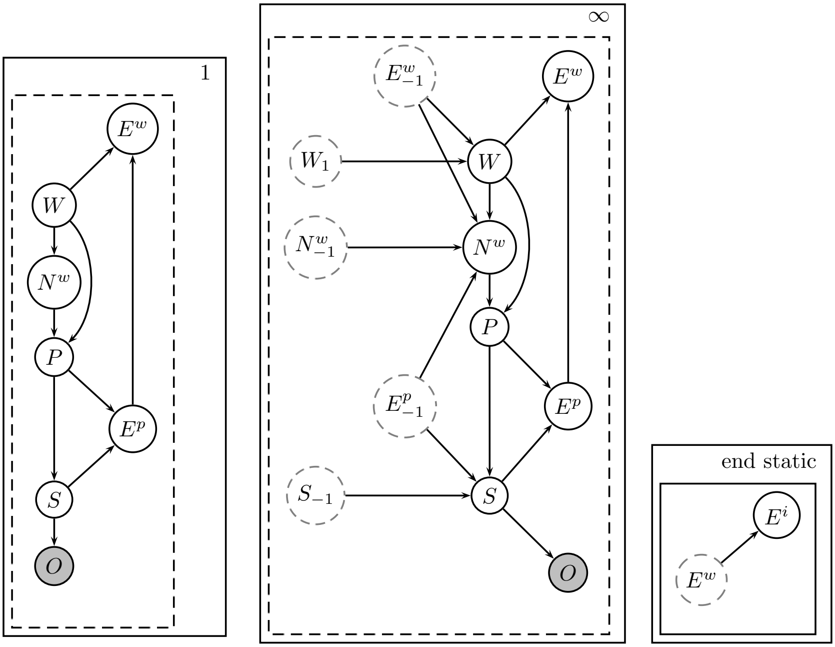


Fig. 1. A speech recognizer implemented with Gaia. O is the Gaussian observation distribution, W represents words, P the phonemes and the S are subphonemic states. The E nodes are binary nodes that signal when a level is allowed to change state.

word is reached if the end of its last phoneme is reached. The end chapter of the model contains a variable E^i which signals the end of the input sequence. It is a child variable of E^w and can only get value 1 if E^w is 1 as well. To make sure that the recognizer will come up with a valid word sequence the value of E^i is clamped to 1. As long a the end of the word is not reached the value of the word variable will remain the same in the next time slice. If the end of the word is reached the next word will be chosen according to a uniform distribution. In the same way, the next subphonemic state is determined.

The original system recognized 77% of all words in the test set correctly, the DBN system using the Gaia inference engine (most probable explanation usign the Frontier algorithm) recognized 75% of all words. The difference in results is due to small differences in the calculations.

6 Conclusion

In this paper we presented our toolkit for speech and language processing with dynamic Bayesian networks and showed how a speech recognizer can be implemented with this toolkit. Recently, we have used the toolkit to implement several language models that include contextual information. In the future we will use it to create a multi-modal large vocabulary speech recognizer that takes contextual information and user characteristics into account.

References

1. Murphy, K.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis, University of California, Berkeley (2002)
2. Huang, T., Koller, D., Malik, J., Ogasawara, G.H., Rao, B., Russell, S.J., Weber, J.: Automatic Symbolic Traffic Scene Analysis Using Belief Networks. In: National Conference on Artificial Intelligence, pp. 966–972 (1994)
3. Singhal, A., Brown, C.: Dynamic Bayes Net Approach to Multimodal Sensor Fusion. In: SPIE Conference on Sensor Fusion and Decentralized Control, Pittsburgh, PA, vol. 3209 (1997)
4. Chella, A., Vitabile, S., Sorbello, R.: A Vision Agent for Mobile Robot Navigation in Time-Variable Environments. In: ICIAP 2001 (2001)
5. Kumagai, T., Akamatsu, M.: Prediction of Human Driving Behavior Using Dynamic Bayesian Networks. IEICE Transactions on Information and Systems E89-D (2006)
6. Perrin, B.E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., d'Alché Buc, F.: Gene Networks Inference Using Dynamic Bayesian Networks. Bioinformatics 19, 138–148 (2003)
7. Zweig, G.: Speech Recognition with Dynamic Bayesian Networks. Ph.D. thesis, Computer Science Division, University of California at Berkeley (1998)
8. Bilmes, J.: Natural Statistical Models for Automatic Speech Recognition. Ph.D. thesis, Dept. of EECS, University of California, Berkeley (1999)
9. Wiggers, P., Rothkrantz, L.J.M.: Topic-Based Language Modeling with Dynamic Bayesian Networks. In: Proceedings of Interspeech 2006 – ICSLP, Pittsburgh, Pennsylvania, pp. 1866–1869 (2006)
10. Livescu, K., Glass, J., Bilmes, J.: Hidden Feature Models for Speech Recognition Using Dynamic Bayesian Networks. In: Proc. Eurospeech (2003)

11. Nefian, A.V., Liang, L., Pi, X., Liu, X., Murphy, K.: Dynamic Bayesian Networks for Audio-Visual Speech Recognition. EURASIP Journal on Applied Signal Processing 11, 1–15 (2002)
12. Wiggers, P., Rothkrantz, L.J.M.: Combining Topic Information and Structure Information in a Dynamic Language Model. In: Text, Speech and Dialogue 2009, pp. 218–225 (2009)
13. Kjaerulff, U.: dhugin: a Computational System for Dynamic Time-Sliced Bayesian Networks. International Journal of Forecasting 11, 89–111 (1995)
14. Murphy, K.P.: The Bayes Net Toolbox for Matlab. Computing Science and Statistics 33, 331–350 (2001)
15. Bilmes, J., Zweig, G.: The Graphical Models Toolkit: An Open Source Software System for Speech and Time-Series Processing. In: Proc. IEEE ICASSP (2002)
16. Druzdzel, M.J.: SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: a Development Environment for Graphical Decision-Theoretic Models. In: AAAI 1999/IAAI 1999, Menlo Park, CA, USA, pp. 902–903 (1999)
17. Madsen, A., Jensen, F.: Lazy Propagation: a Junction Tree Inference Algorithm Based on Lazy Evaluation. Artificial Intelligence 113, 203–245 (1999)
18. Hulst, J.: Modeling Physiological Processes with Dynamic Bayesian Networks. Master's thesis, Man-Machine Interaction Group, Delft University of Technology (2006)
19. Zweig, G., Padmanabhan, M.: Exact Alpha-Beta Computation in Logarithmic Space with Application to Map Word Graph Construction. In: Proceedings of ICSLP 2000, Beijing, China (2000)
20. Damhuis, M., Boogaart, T., In 't Veld, C., Versteijlen, M., Schelvis, W., Bos, L., Boves, L.: Creation and Analysis of the Dutch Polyphone Corpus. In: Proceedings of ICSLP 1994, Yokohama, Japan, pp. 1803–1806 (1994)
21. Young, S., Evermann, G., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.2.1) (2002)

Special Speech Synthesis for Social Network Websites

Csaba Zainkó, Tamás Gábor Csapó, and Géza Németh

Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Hungary
{zainko,csapot,nemeth}@tmit.bme.hu

Abstract. This paper gives an overview of the design concepts and implementation of a Hungarian microblog reading system. Speech synthesis of such special text requires some special components. First, an efficient diacritic reconstruction algorithm was applied. The accuracy of a former dictionary-based method was improved by machine learning to handle ambiguous cases properly. Second, an unlimited domain text-to-speech synthesizer was applied with extensions for emotional and spontaneous styles. Chat or blog texts often contain ”emoticons” which mark the emotional state of the user. Therefore, an expressive speech synthesis method was adapted to a corpus-based synthesizer. Four emotions were generated and evaluated in a listening test: neutral, happy, angry and sad. The results of the experiments showed that happy and sad emotions can be generated with this algorithm, with best accuracy for female voice.

Keywords: diacritic restoration, emotional speech synthesis, microblog reading system, chat-to-speech.

1 Introduction

This paper gives an overview of the design concepts and implementation steps of a Hungarian microblog text-to-speech reading system. Microblog websites (e.g. Twitter, http://twitter.com) and chat-like talking applications are very popular nowadays. In chat applications, where little talk is written, it is advantageous to use speech instead of always keeping track of the dialog. The user can do something else than looking at the screen, and he will still know what is being said in the chat channel. This scenario is mainly useful when messages do not arrive very often. A microblog reader system can also be useful in mobile environment, because there is no possibility to continuously watch the display or the user does not have a free hand to handle the device (e.g. during car driving, or sport activities like running). Another possible situation is if the user is working with a full screen desktop application and he needs real time information from social networks. Loud reading demands only short time attention, and task changing is not necessary. This system is very useful for visually impaired and blind people, as well.

However, chat-to-speech synthesis conveys some new problems. In current web-based social networks people tend to use the special form of their language (e.g. letters without diacritics for Hungarian and with ”emoticons”). Text repairing algorithms are needed to recover the proper text that can be read by a TTS. Spontaneous style and emotional synthesized speech can help to improve how people accept these systems. There exist several systems that are specialized in chat reading. For example, an attempt

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 455–463, 2010.

has been made to fit together the free Espeak utility with the X-Chat IRC client [1]. A couple of other applications exists which use a TTS to read any type of documents, including web pages and blog sites, but most of them are prepared without fitting a general TTS to the specific task of chat reading.

Our approach is a first step in developing a Hungarian microblog reading system with more complex functions. Section 2 introduces the problem of missing diacritics and a combined machine learning approach to solve it. Section 3 discusses several approaches to make synthesized speech more spontaneous and thus more human-like. Section 4 shows a method to transform neutral speech to emotional. The last section summarizes and concludes the paper.

2 Diacritics Restoration

Characters with diacritics occur in large numbers in most European languages. According to [2], only the English alphabet is without diacritics from 36 languages studied. In several telecommunication applications, like SMS in cell phones, some or all diacritics of the input text are removed because of character encoding. On many small devices the typing of diacritic letters is uncomfortable and slow so people tend to use the diacriticless letters of their language when writing computer documents and Web 2.0 websites. This is a hard problem for a text-to-speech system: the errors in the spoken utterances are much more confusing than in written text. Therefore, we apply a diacritic restoration algorithm which can formulate proper input text for a speech synthesizer.

The Hungarian language has five ambiguous sets of letters from the viewpoint of diacritics. These include nine letters with a diacritic, all of which are vowels. Some of the ambiguous sets differ only in quantity ("i-í", "o-ó", "ö-ő", "u-ú", "ü-ű"), while the others may differ in quantity and vowel quality as well ("a-á", "e-é", "o/ó-ö/ő", "u/ú-ü/ű").

Diacritic restoration is a well-known problem, and there are several methods to solve this task. With dictionary-based solutions up to 90% accuracy was reported in accent restoration depending on the language [3]. The use of Hidden Markov Models can lead to almost perfect restoration performance [4]. Most word based methods need a reliable morphological analysis tool. On the other hand, letter based methods are much easier to build and provide generalization beyond words [2].

A former dictionary-based algorithm for Hungarian selected always the word variant with the most possible diacritic pattern [5]. This method gives poor results when the variants occur nearly equally often. As this solution does not have generalization capability, words that are not included in the trainig corpus cannot be handled properly.

2.1 Combined Diacritics Restoration Method

We use a combined method applying a word level dictionary and a letter level machine learning approach together. The unambiguous cases (words with only one possible diacritic pattern, e.g. *az*=*the* is a Hungarian article, but *áz* does not occur as a Hungarian word) are handled using a dictionary, while the diacritics of the ambiguous cases (words with more than one possible diacritic patterns, e.g. *meg*-*még*=*plus*-*still*) are calculated using a decision tree.

First, a learning phase is performed. As training data, the Hungarian National Corpus (HNC) of 187 million words is applied [6]. HNC is a collection of written linguistic data representing present-day standard Hungarian. There are 3.58 million different word forms in the corpus (880 thousand without any diacritic letter and 2.7 million with at least one diacritic letter).

The first step of the learning phase is the separation of ambiguous and unambiguous words from the viewpoint of diacritics. For this, we use a naive algorithm: those words are marked as unambiguous, which have only one diacritic pattern variant, and the rest build up the ambiguous set of words. Those words are included in the dictionary as well, which have one diacritic pattern in more than 95% of the cases. These unambiguous cases cover about 84.5% of the input text. To handle typos, we applied a spell checker only on the ambiguous part, because on the unambiguous part the spell checker threw out too many valuable correct forms. The ambiguous part (15.5% of the corpus) is handled with a J4.8 decision tree [7], which is the open-source and improved implementation of the popular C4.5 decision tree. This learning phase was conducted using the Rapidminer data mining program (`http://www.rapidminer.com`). The ambiguous set of the input data is about 29 million cases. The parameters of the J4.8 tree were optimized in order to get an acceptable-sized tree that can be handled in an application. In order to improve accuracy of the diacritic restoration, the 100 most frequent ambiguous words (covering 60 % of wrong decisions of the former method in the ambiguous set) were trained with separate decision trees for each. During the decision tree learning phase, the context of 20 letters of the ambiguous vowels is extracted as the training data. This is twice the window as suggested by [2], but we intended to treat correctly very long words as well.

After the learning phase, the missing diacritics of the input text can be determined using the above-mentioned algorithm. First, the input text is separated into unambiguous and ambiguous words. The diacritized versions of the unambiguous words are searched in the dictionary. The context of the ambiguous words is calculated from the input text. After that, the J4.8 decision tree is applied and the restored diacritics are given as the output text.

2.2 Accuracy of Diacritic Restoration

Word accuracies were calculated because for the TTS domain correct words are more important than reconstructed vowels alone. Partly incorrect diacritic reconstruction degrades the quality of speech.

With our combined method, we applied training and validation on the “DIA”, “Personal” sub-corpora and the whole HNC database. Accuracies of 97.7% for “DIA”, 97.2% for “Personal” and 98.2% for the whole HNC can be reached in diacritic restoration applied to the Hungarian language. Detailed sub-results for the ambiguous cases only are shown in Table 1.

The three rows are the different test sets. The first (“DIA”) is the Digital Literature Academy which was used in [2] as well. The second (“Personal”) is a collection of the web forum sub-corpus of HNC, which is the most similar in topic to our target application. The third is the whole HNC database. The “Baseline amb.” column gives the results of diacritics restoration applied to ambiguous cases when the decision is

Table 1. Results of diacritic restoration for ambiguous cases

	Baseline amb.	Top100 amb.	Ambiguous
DIA (DLA)	75.9%	92.7%	82.4%
Personal	71.4%	88.5%	80.8%
HNC (all)	75.2%	92.9%	82.9%

always the most likely form, as in the algorithm introduced in [5]. The next column shows the results of the 100 separately trained words. The third column gives the results of the rest of words that are known as ambiguous and are not contained in the previous column.

The tendency of percentages is similar in all of the test sets. The most relevant test set is the "Personal" for us. The word accuracy increased from 71.4% to 88.5% with 100 separately trained words and to 80.8% with the other ambiguous words. The trained decision tree can restore correct diacritics in 70.2% of the general cases (not included in Table 1). This number is an estimate for the accuracy of the solution for the out of dictionary words.

3 Spontaneous Synthesized Speech

Spontaneous speech is the most natural oral expression of humans. However, speech synthesis has mainly focused on read speech (e.g. in the form of huge read corpora) because it is much easier to process than spontaneous speech. According to [8], it is advantageous to mimic some aspects of spontaneous speech in a TTS system. This style is particularly useful in our chat and microblog reading system. It should be chosen, which properties of spontaneous speech are worth to be modelled in a human-machine communication system. For example, hesitation and humming are significant attributes of everyday speech, but they increase the cognitive load of the listener, thus disturb the understanding of a speaking machine system.

3.1 Corpus-Based Unit Selection TTS

The unit selection TTS that was used in our experiments is described in detail in [9]. The currently used speech databases contain sentences from several domains (e.g. weather forecasts and radio news). The synthesizer can generate the prosody in two ways, depending on the type of the input sentence. If the sentence fits in the domain of the corpus, a simple prosody model is used, based on the relative position of words within a prosodic phrase. Because it is based on words, it will work properly only if most words of the input sentence are found in the corpus. If the sentence is out of theme, there will not be enough whole words, which can determine the prosody. On those parts of the sentences a template-based F_0 generation method [10] is applied. The templates are based on spontaneous speech. The obtained F_0 values are used in the target cost function of the TTS to follow the F_0 curve. This extension can help to use the originally limited domain TTS in the unlimited domain of chat and microblog reading.

3.2 Spontaneous Style Synthesized Speech

Several spontaneous like synthesized speech examples are generated, in the form of modifying the output of the corpus-based speech synthesizer. These utterances included some properties of spontaneous style speech. First, filled pauses were added, in the form of breathing, after the conjunctive words. Second, silent pauses were lengthened to mimic “thinking” during everyday speech. Third, hesitation and humming were added randomly and the F_0 curve was shifted. At last, the structure of the sentences (e.g. word order) was modified in order to come closer to spontaneous speech.

3.3 Listening Test and Results

Two sentences were chosen for evaluation in a listening test. The variants of the sentences are shown in Table 2. The first variant was produced with a diphone system using copied natural prosody. Variants 2–5 were generated with the corpus-based unit selection TTS. The second variant contained the sentence with a re-edited structure. In the third variant, we applied the baseline position-based prosody of the TTS. In variants 4 and 5 hesitation, silent and filled pauses were added and the F_0 contour was modified to approximate spontaneous speech. The last variant was a natural spontaneous speech sample.

A small web-based listening test was conducted in order to get feedback from speech scientists. The goal of the test was to investigate, how people accept the modelled properties of spontaneous speech. After listening to each of the 6–6 variants of the two sentences, the listeners had to answer two 5-point MOS questions: Q1) “To what extent is this speech sample natural?” 5 – very human-like, . . . 1 – very machine-like; Q2) “To what extent is this speech sample spontaneous?” 5 – totally spontaneous, . . . 1 – absolutely not spontaneous.

Seven listeners evaluated the sentences (all of them were Hungarian phoneticians or speech synthesis experts; 3 male; 4 female; mean age: 33; mean test duration: 4 minutes). The results are included in Table 2. According to the results, the insertion of breathing and the longer pauses helped to approximate the spontaneous speech, but the quality of speech was decreased. Despite of the frequent occurrence of hesitation in human speech, listeners of the test did not prefer the synthesized sentences with inserted hesitation. The diphone-based variant achieved very high spontaneous score, because its prosody was copied from the natural sample, but its quality is lower than that of unit selection TTS samples.

This simple approach showed that it is possible to model several properties of spontaneous speech in a TTS system. In a real application, users tend to accept only quiet breathing and some laughter. Several sentences should be read together in one prosodic phrase, as in spontaneous speech we also talk in longer units.

4 Emotional Synthesized Speech

Usually there are few coherent sentences in a microblog, therefore the reader or listener may misunderstand the content without emotional signs. During writing the bloggers

Table 2. Variants of speech samples used in the listening test and their MOS results

			Sent. 1		Sent. 2	
Num	Technology	Type	Q1	Q2	Q1	Q2
1	Diphone	Rule based prosody	2.00	3.86	2.25	4.38
2	Corpus	Reedited sentence structure	2.50	3.00	2.38	3.00
3	Corpus	Position based prosody	2.13	2.38	2.13	2.75
4	Corpus	Added hesitation	2.13	3.38	2.88	3.50
5	Corpus	Added hesitation, pauses; modified F_0	2.63	2.88	2.88	3.50
6	Human	Natural	5.00	5.00	5.00	5.00

use ”emoticons”, but these cannot be read in themselves. These emoticons modify the meaning of the previous word or sentence. During speech synthesis it is beneficial to modify the natural sentences to their proper emotional sentences. In order to be able to quickly identify the emotional type of the heard sentence we use four emotions: neutral style and angry :@, happy :), sad :(. The ”Personal” subcorpus of HNC contains 90 thousand emoticons (in 1.5 million sentences).

4.1 Emotion Modification Algorithm

The emotion modification method was chosen considering many points of view. Bulut et. al. [11] emphasize that for creating strong emotions (like angry and happy), the segmental components of speech are important features. Prosody is also important, but if the segmental structure is set properly, the emotion will be identifiable with a less correct prosody as well. It can be used in such speech synthesis technologies where prosody modification is limited or not allowed (e. g. the applied corpus based unit selection system). The modification method should be applied on the output synthesized speech signal because this corpus-based TTS does not encourage strong signal modification.

Přibilová and Přibil [12] describe a method which is based on spectrum modification. A non-linear frequency scale transformation is applied on the speech spectral envelope. The main suprasegmental features are also modified: F_0, energy and duration. During human emotional speech, individual formants are shifted. This is caused by the physiological change of the vocal tract, and the distribution of low and high-frequency energy is also changing. This is the concept of spectral transformation used in emotion modification.

Our spectral transformation method is based on the PSOLA algorithm and uses non-linear frequency scaling as suggested by [12]. First, the pitch markers are determined by the Praat program or the synthesis system computes them. An asymmetric Hann window function was used. The center of the window is at the pitch marks and the left and right end of the window are at the previous and next pitch marks. This time domain windowed signal is converted to frequency domain with the DFT (Discrete Fourier Transform) algorithm. The amplitude part is modified with a transform function [12] and the low-frequency and high-frequency energy distribution is set properly. The modification of the phase spectrum is not necessary. With an inverse DFT algorithm we convert back the signal to time domain. Before finishing the speech output, a second Hann window corrects the left and right ends of the signal to remove discontinuities. The segments are recombined with F_0 and duration modifications similar to the original PSOLA algorithm.

4.2 Experiments

Several emotional variants of three sentences with a male and a female voice were tested in a listening experiment in order to verify our hypothesis that the spectrum modification can improve emotion transformation. One sentence was the modification of a natural speech sample uttered by a professional female speaker, while the other two sentences were the modifications of the output of the corpus based speech synthesizer with a female and a male voice. The modifications included the methods described in Sec 4.1.

The modification parameters were similar to [12]. γ_1 and γ_2 are varied in the sentences with ± 0.05. F_0 is increased by 15% for anger, 17% for happy and decreased by 16% for sadness. Energy is increased by 4.6 dB for anger, 2.3 dB for happy and decreased by 3dB for sadness. The spectral energy distribution is also modified. For sadness the low-frequency components are increased by 4 dB and for the other two emotions the high-frequency components are increased by 2 dB for happy and 4 dB for angry. The cut-off frequency was 1 kHz.

A separate listening test was conducted to determine the perceived emotion, naturalness and quality of synthetic sentences. The test was web-based, in order to emulate the circumstances of a potential application. 25 native speakers of Hungarian participated in the test with no known hearing loss. The results of 3 listeners were excluded from the evaluation because they either did not finish the test, or were found to respond randomly. The remaining 22 listeners consisted of 15 male and 7 female testers having a mean age of 38 years. 8 listeners used head- or earphones while 14 testers listened to loudspeakers. The listening test took 9.6 minutes to complete, on average.

The listeners had the option to replay a stimulus as many times as they wished, but they were not allowed to go back to a preceding stimulus, once they rated it. The playback order was randomized individually for each listener.

4.3 Results of the Listening Test

The confusion matrix of the planned and recognized emotions is shown in Table 3, for the three sentences separately. From the 3–3 parameter variants of the sentences, the 1–1 best were selected from the viewpoint of highest recognition of intended emotion. Only these are included in the table.

With the female and male voice TTS, the sad emotion could be produced with the highest accuracy. In the female case, the happy emotion is acceptable as well, while in the male case, listeners mostly misrecognized the happy variants. The angry emotion could be generated better with the male TTS. Emotion modification caused high accuracies in happy and sad natural speech, while the angry emotion was less successful. Přibilová and Přibil [12] report on similar results, the worst identification is for angry, and sadness gets the best scores.

Table 3. Confusion matrix results of the emotion recognition

		Recognized														
		N	A	H	S		N	A	H	S		N	A	H	S	
	N	82%	0%	14%	5%	N	45%	23%	0%	27%	N	77%	0%	18%	5%	N=neutral
Planned	A	27%	27%	41%	5%	A	36%	41%	0%	23%	A	45%	32%	14%	9%	A=angry
	H	27%	5%	68%	0%	H	41%	23%	14%	23%	H	9%	5%	82%	5%	H=happy
	S	9%	5%	0%	86%	S	14%	5%	0%	81%	S	23%	5%	5%	68%	S=sad
		TTS-female					TTS-male					Natural-female				

According to the comments of the listeners, the emotional modification decreased the quality of speech only in the angry case. They reported that the utterances were more natural for the female sentences. This can be explained by a note of [12]: the parameters of the applied spectral modification algorithm were worked out for female speech. This test showed that for male speech, parameters should be applied in some other range.

5 Summary

In this paper, the concepts of a Hungarian chat and microblog reading speech synthesis system were introduced. We extended a former dictionary-based diacritic restoration algorithm with machine-learning applied at letter level, getting a combined method. This approach can lead to 98.2% accuracy for general topic input text and to 97.2% accuracy on the specific text of forum entries.

A limited domain corpus-based text-to-speech synthesizer was extended for use in the specific task of chat reading. Several properties of spontaneous speech were investigated. Some of them were integrated in the TTS system in order to create "loose style" machine-generated speech that is closer to that of used in our everyday human-human communication. The output speech of the synthesizer was passed into a spectral modification algorithm in order to produce emotional speech. Four emotions were investigated, of which the female version of happy and sad could be synthesized in the best quality according to a listening test.

This paper is an exploratory study in developing a specialized speech synthesis system. The proposed method can be applied in a chat reading program or with any social network web-site (e.g. Twitter). Our future plans include the completion of such a system.

Acknowledgments. We thank the listeners for participating in the listening tests. This research was supported by the TELEAUTO (OM-00102/2007) project of the Hungarian National Office for Research and Technology and by the ETOCOM project (TÁMOP-4.2.2-08/1/KMR-2008-0007).

References

1. X-Chat Text-To-Speech, `https://launchpad.net/xctts`
2. Mihalcea, R., Nastase, V.: Letter Level Learning for Language Independent Diacrtitics Restoration. In: COLING 2002, Taipei, Taiwan, pp. 1–7 (2002)

3. Galicia-Haro, S.N., Bolshakov, I.A., Gelbukh, A.F.: A Simple Spanish Part of Speech Tagger for Detection and Correction of Accentuation Error. In: Matoušek, V., Mautner, P., Ocelíková, J., Sojka, P. (eds.) TSD 1999. LNCS (LNAI), vol. 1692, pp. 219–222. Springer, Heidelberg (1999)
4. Simard, M.: Automatic Insertion of Accents in French Text. In: Proc. of Conf. EMNLP, Granada, Spain, pp. 27–35 (1998)
5. Németh, G., Zainkó, C., Fekete, L., Olaszy, G., Endrédi, G., Olaszi, P., Kiss, G., Kiss, P.: The Design, Implementation and Operation of a Hungarian E-Mail Reader. Int. Journ. Of Speech Techn. 3-4, 216–228 (2000)
6. Hungarian National Corpus, http://corpus.nytud.hu/mnsz
7. Witten, I.H., Frank, E.: Using the J4.8 Decision Tree, Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
8. Carlson, R., Gustafson, K., Strangert, E.: Modelling Hesitation for Synthesis of Spontaneous Speech. In: Proc. of Speech Prosody, Dresden, pp. 69–72 (2006)
9. Fék, M., Pesti, P., Németh, G., Zainkó, Cs., Olaszy, G.: Corpus-Based Unit Selection TTS for Hungarian. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 367–374. Springer, Heidelberg (2006)
10. Csapó, T.G., Zainkó, Cs., Németh, G.: A Study of Prosodic Variability Methods in a Corpus-Based Unit Selection Text-To-Speech System. Infocommunications Journal LXV, 32–37 (2010)
11. Bulut, M., Narayanan, S. S., Syrdal, A.K.: Expressive Speech Synthesis Using a Concatenative Synthesizer. In: Proc. ICSLP, pp. 1265–1268 (2002)
12. Přibilová, A., Přibil, J.: Spectrum Modification for Emotional Speech Synthesis. Multimodal Signals: Cognitive and Algorithmic Issues, 232–241 (2009)

Robust Statistic Estimates for Adaptation in the Task of Speech Recognition*

Zbyněk Zajíc, Lukáš Machlica, and Luděk Müller

University of West Bohemia in Pilsen,
Faculty of Applied Sciences, Department of Cybernetics,
Univerzitní 22, 306 14 Pilsen
zzajic@kky.zcu.cz, machlica@kky.zcu.cz, muller@kky.zcu.cz

Abstract. This paper deals with robust estimations of data statistics used for the adaptation. The statistics are accumulated before the adaptation process from available adaptation data. In general, only small amount of adaptation data is assumed. These data are often corrupted by noise, channel, they do not contain only clean speech. Also, when training Hidden Markov Models (HMM) several assumptions are made that could not have been fulfilled in the praxis, etc. Therefore, we described several techniques that aim to make the adaptation as robust as possible in order to increase the accuracy of the adapted system. One of the methods consists in initialization of the adaptation statistics in order to prevent ill-conditioned transformation matrices. Another problem arises when an acoustic feature is assigned to an improper HMM state even if the reference transcription is available. Such situations can occur because of the forced alignment process used to align frames to states. Thus, it is quite handy to accumulate data statistic utilizing only reliable frames (in the sense of data likelihood). We are focusing on Maximum Likelihood Linear Transformations and the experiments were performed utilizing the feature Maximum Likelihood Linear Regression (fMLLR). Experiments are aimed to describe the behavior of the system extended by proposed methods.

Keywords: fMLLR, adaptation, speech recognition, robustness.

1 Introduction

Nowadays, the adaptation of an acoustic model is used as a standard tool to improve the performance of Hidden Markov Model (HMM) in the task of speech recognition. In real conditions still several complications should be solved. The main problem arises in cases with low amount of adaptation data, even more if their quality is in question (noise, channel, etc.) [1]. Our effort is to decrease the vulnerability of the system in mentioned conditions.

In order to handle low amount of data the fMLLR adaptation is commonly utilized. Since fMLLR uses clustering of similar model parameters it suffices only with small

[*] This research was supported by the Ministry of Education of the Czech Republic, project No. MŠMT LC536, the Grant Agency of the Czech Republic, project No. GAČR 102/08/0707 and the grant of The University of West Bohemia, project No. SGS-2010-054.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 464–471, 2010.

amount of observation vectors. fMLLR approach is addressed in Section 2.2. However, in cases of on-line speech recognition [2], when data statistics are accumulated continuously, also fMLLR approach can be faced with difficulties in the sense of ill-conditioned transformation matrices.

The estimation of fMLLR matrices is an iterative procedure usually initialized with random matrices. Therefore, some procedures were developed to initialize fMLLR in order to ensure the stability of estimates of transformation matrices. One of the methods is described in Section 3.1. Another problem concerns inaccuracies in the phone alignment to HMM states caused by imperfect model training, and/or problematic data. To avoid the problem of incorrect model shifting via adaptation we are trying to discard non informative mixture components from the process of statistics accumulation (see Section 2.1). This approach, named Refinement of statistics, can be found in Section 3.2. Experiments were carried out utilizing two distinct corpora, one was used to adjust refinement values defined in Section 3.2, and the second was used to test the validity of these values, see Section 4. The article aims to describe the behavior of proposed methods, the discussion of results can be found in Section 4.4.

2 Adaptation Techniques

The task of adaptation is to shift the unadapted model in the direction of new (adaptation) data. The unadapted model is often denoted as Speaker Independent (SI) model. We will focus on HMMs with output probabilities of states represented by GMMs. GMM of the $j-th$ state is characterized by a set $\boldsymbol{\lambda}_j = \{\omega_{jm}, \boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm}\}_{m=1}^{M_j}$, where M_j is the number of mixture components, ω_{jm}, $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{C}_{jm}$ are weight, mean and variance of the $m-th$ mixtures' component, respectively. Well know adaptation techniques are Maximum A-posteriori Probability (MAP) [3] and Linear Transformations based on the Maximum Likelihood [4].

2.1 Statistics of Adaptation Data

The adaptation techniques do not access the data directly, but only through some statistics defined as:

$$\gamma_{jm}(t) = \frac{\omega_{jm} p(\boldsymbol{o}(t)|jm)}{\sum_{m=1}^{M} \omega_{jm} p(\boldsymbol{o}(t)|jm)} \tag{1}$$

stands for the posterior of the $j-th$ state and the $m-th$ mixtures' component of the HMM. It should be noted that the pertinence of the feature vector $\boldsymbol{o}(t)$ to the $j-th$ state is given by the forced alignment process utilizing the reference transcription. Next,

$$c_{jm} = \sum_{t=1}^{T} \gamma_{jm}(t) \tag{2}$$

is the soft count of mixture component m,

$$\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)\boldsymbol{o}(t)}{\sum_{t=1}^{T} \gamma_{jm}(t)}\ , \quad \boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)\boldsymbol{o}(t)\boldsymbol{o}(t)^{\mathrm{T}}}{\sum_{t=1}^{T} \gamma_{jm}(t)} \tag{3}$$

represent the first and the second moment of features which align to mixture component m in the j-th state of the HMM. Note that $\boldsymbol{\sigma}_{jm}^2$ = diag($\boldsymbol{C}_{jm}$) is the diagonal of the covariance matrix $\boldsymbol{C}_{jm}$.

2.2 Feature Maximum Likelihood Linear Regression (fMLLR)

This technique belongs to the category of Linear Transformations (LTs), another LT based method is Maximum Likelihood Linear Regression (MLLR). These methods utilizes clustering of similar model components [6], thus clusters $K_n, n = 1, \ldots, N$ are formed. Hence, lower amount of adaptation data is needed to update the model. The fMLLR is used to transform directly features $\boldsymbol{o}(t)$ according to

$$\bar{\boldsymbol{o}}_t = \boldsymbol{A}_{(n)}\boldsymbol{o}_t + \boldsymbol{b}_{(n)} = \boldsymbol{W}_{(n)}\boldsymbol{\xi}(t) \;, \tag{4}$$

where

$$\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}], \tag{5}$$

$\boldsymbol{W}_{(n)}$ represents the transformation matrix corresponding to the $n - th$ cluster K_n and $\boldsymbol{\xi}(t) = [\boldsymbol{o}_t^{\mathrm{T}}, 1]^{\mathrm{T}}$ stands for the extended feature vector. The auxiliary function can be written in the form

$$Q_{\boldsymbol{W}_{(n)}}(\boldsymbol{\lambda}, \bar{\boldsymbol{\lambda}}) = \log|\boldsymbol{A}_{(n)}| - \sum_{i=1}^{I} \boldsymbol{w}_{(n)i}^{\mathrm{T}}\boldsymbol{k}_i - 0.5\boldsymbol{w}_{(n)i}^{\mathrm{T}}\boldsymbol{G}_{(n)i}\boldsymbol{w}_{(n)i} \;, \tag{6}$$

where

$$\boldsymbol{k}_{(n)i} = \sum_{m \in K_n} \frac{c_m \mu_{mi} \quad {}_m(\;\;)}{\sigma_{mi}^2} \;, \quad \boldsymbol{G}_{(n)i} = \sum_{m \in K_n} \frac{c_m \quad {}_m(\quad {}^{\mathrm{T}})}{\sigma_{mi}^2} \tag{7}$$

and

$$\boldsymbol{\varepsilon}_m(\boldsymbol{\xi}) = \left[\boldsymbol{\varepsilon}_m^{\mathrm{T}}(\boldsymbol{o}), 1\right]^{\mathrm{T}} \;, \quad \boldsymbol{\varepsilon}_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}}) = \begin{bmatrix} \boldsymbol{\varepsilon}_m(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) & \boldsymbol{\varepsilon}_m(\boldsymbol{o}) \\ \boldsymbol{\varepsilon}_m^{\mathrm{T}}(\boldsymbol{o}) & 1 \end{bmatrix} . \tag{8}$$

The solution of the minimization auxiliary function (6) can be found in [5]. Matrices $\boldsymbol{A}_{(n)}$ and $\boldsymbol{b}_{(n)}$ are estimated iteratively. Thus, they have to be initialized, e.g. as a diagonal matrix with ones on the diagonal and a zero vector, respectively.

3 Robustness

The task is to design the estimation of adaptation formulas as robust as possible in order to increase the systems efficiency in problematic situations - small amount of adaptation data, noise, model inaccuracies, etc. Such problems are handled in following subsections.

3.1 Inicialization of Adaptation Matrix

The auxiliary matrices (7) are dense and have a lot of parameters to be estimated. One of the problem arises in cases when low amount of adaptation data is available. Such situations can lead to ill-conditioned transformation matrices (5) and degradation of systems' performance. Therefore it is suitable to initialize matrices (7) with proper values in order to increase the robustness of the estimation process. The idea is to utilize data that fits the model to be adapted (when none new adaptation data are available, the estimated transformation matrix should equal the identity matrix). However, as already mentioned in Section 2.1 we do not need the data directly, we need only their statistics (mean and variance). Thus, we can use directly the unadapted model parameters as proposed in [8]. Now the initialization of (7) takes the form

$$\boldsymbol{k}_{(n)i} = \sum_{m \in K_n} p_m \frac{\mu_{mi}}{\sigma_{mi}^2} \begin{bmatrix} \boldsymbol{\mu}_m \\ 1 \end{bmatrix}, \; \boldsymbol{G}_{(n)i} = \sum_{m \in K_n} p_m \frac{1}{\sigma_{mi}^2} \begin{bmatrix} \boldsymbol{\mu}_m \boldsymbol{\mu}_m^T + \boldsymbol{C}_m & \boldsymbol{\mu}_m^T \\ \boldsymbol{\mu}_m & 1 \end{bmatrix}, \qquad (9)$$

where $\boldsymbol{\mu}_m$,$\boldsymbol{C}_m$ are parameters of the m-th mixture component of the SI model belonging to the cluster $K_{(n)}$, p_m is a smoothing weight. Greater values of p_m indicate greater influence of the initialization, but the adaptation is less effective since the estimates are more restricted. In our case we set p_m equal to the weight of a mixture component, $p_m = \omega_m$. Other initialization approaches were studied in [9].

3.2 Refinement of Statistics

In order to accumulate the statistics (see Section 2.1) a proper phone alignment to HMM states is required. Even if a reference transcription is available the forced alignment can contain errors. These are caused among others by improper assumptions of the suitability of the HMM, e.g. the Maximum Likelihood (ML) training may not yield the most appropriate estimates [7].

We have investigated several approaches how to restrict the selection of mixture components used for estimation of transformation matrices. One of the possibilities is to discard a whole mixture component from a cluster (mixture component does not participate in (7)) based on its soft count (2). However, for some correctly aligned features the discarded component could be crucial (in the sense of its involvement in the estimation of G and k in (7)), but for some features such mixture component introduces only inaccuracies. Therefore it is more convenient to judge the components' suitability for each feature according to components' posterior defined in (1), and not reflect increments in statistic for inconvenient mixture components with low posteriors for a given feature (equivalent to setting the posterior $\gamma_{jm}(t)$ to zero in (7)).

Two possibilities were studied how to discard feature statistics according to the posterior $\gamma_{jm}(t)$. One could assign a threshold Th_γ and take into account only statistics related to mixture components with posteriors higher than the threshold Th_γ. Hence, $\gamma_{jm}(t) = 0$ if the inequality $\gamma_{jm}(t) \geq Th_\gamma$ is not met. Such an approach reflects an assumption that two hypothesis H_0, H_1 overlap, where H_1: *feature o_t was generated by mixture component m*, H_0: *feature o_t was NOT generated by mixture component m*, and we want to minimize the incorrect rejection of hypothesis H_0. A different approach accumulates only statistics acquired for N-best mixture components (with respect to their posterior) in one HMM state. These two techniques can be also combined - at first N-best mixture components are chosen and then the threshold Th_γ is applied to further exclude inconvenient components.

4 Experiments

We have utilized two data sets to test the systems performance. First set was used for estimation of refinement parameters and for some additional experiments related to amount of adaptation data and the second set was used to confirm the validity of estimated refinement values.

4.1 Czech Telephone (CzT) Corpus

Corpus consists of Czech read speech transmitted over a telephone channel. The digitization of an input analog telephone signal was provided by a telephone interface board DIALOGIC D/21D at 8 kHz sample rate and converted to the mu-law 8 bit resolution. The corpus was divided into two parts, the training set and the testing set. The training set consisted of 100 speakers, where each of them read 40 different sentences (length of each sentence was cca 5 sec.). The testing set consisted of 100 speakers not included in the training set, where each of them read the same 20 sentences as the other, further divided into two groups. The first one contained 15 sentences used as adaptation data and the second one contained 5 remaining sentences used for testing of adapted models. The vocabulary in all our test tasks contained 1,260 different words. There were no OOV (Out Of Vocabulary) words. The basic speech unit of our system is a triphone. Each individual triphone is represented by a three states HMM; each state is provided by 8 mixtures of multivariate Gaussians. We are considering just diagonal covariance matrices. In all recognition experiments a language model based on zerograms was applied. It means that each word in the vocabulary is equally probable as a next word in the recognized utterance.

4.2 SpeechDat-East (SD-E) Corpus

SpeechDat-East [10] contains telephone speech in 5 languages Czech, Polish, Slovak, Hungarian, and Russian. For our experiments data were chosen in a similar fashion as CzT. We used only the Czech part of SD-E. The acoustic HMM was trained on 700 speakers with 50 sentences for each speaker (cca 5 sec. for sentence). For testing purposes 150 speakers were chosen with 50 sentences for each speaker. The vocabulary consisted of 7,000 words. No OOV words were present. Triphones were model using 3 state HMM with 8 gaussian mixtures (diagonal covariances) in each of the states. For the recognition a language model based on zerograms was considered.

4.3 Adaptation Setup

In our experiments we utilized fMLLR adaptation with clustering performed via the Regression Tree (RT). RT was constructed using the HTK toolkit [11]. The threshold for occupation of nodes in the regression tree was set to 1,000. Thus, approximately 10–15 matrices for one speaker were computed. Instead of the model directly the features were transformed. Only one iteration of fMLLR was carried out.

4.4 Results

First experiments involved the CzT set (see Section 4.1). Initially, we tried to find (empirically) the value of the threshold Th_γ for posterior $\gamma_{jm}(t)$ (see Section 3.2). Results of refined fMLLR (r-fMLLR) and combination of refined and initialized fMLLR (ri-fMLLR) can be found in Figure 1. As can be seen the best performance is achieved around the value $Th_\gamma = 0.5$ and a gain of 0.21% absolutely is acquired (see Table 1).

Instead of the threshold Th_γ one can choose N-best mixture components (according to their posterior $\gamma_{jm}(t)$ (1)) for statistics accumulation. Results of the N-best components approach together with results for $Th_\gamma = 0.5$ and basic fMLLR can be found in Table 1. These two refinement methods give very similar results. Basically, Th_γ controls the number of involved mixture components and so does the N-best approach. If Th_γ is set high enough only the best mixture component is chosen (though sometimes none) and the method works almost identically to the 1-best approach. The decrease of Th_γ is comparable to the increase of N. However, Th_γ balances the number of mixture components for each feature vector. Thus, in the rest of the paper we utilize only $Th_\gamma = 0.5$.

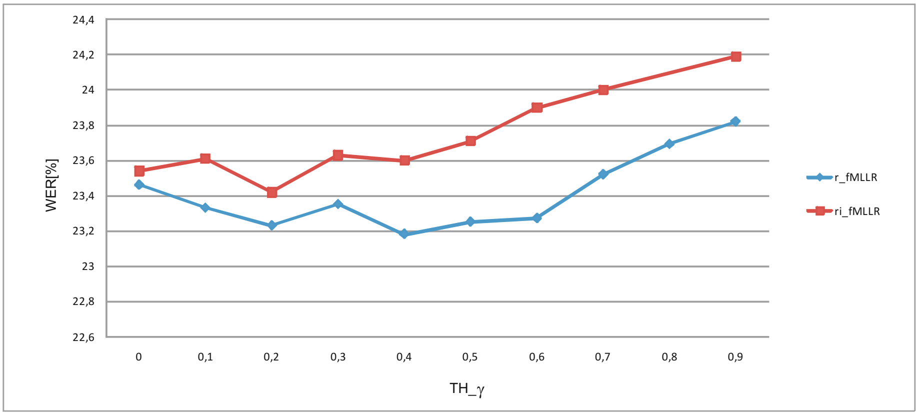


Fig. 1. Czech Telephone Corpus. Word Error Rate (WER)[%] of systems based on fMLLR adaptation with dependency on Th_γ, where r-fMLLR and ri-fMLLR denotes refined fMLLR and refined initialized fMLLR, respectively.

We have also investigated the dependency of the adaptation performance on the amount of adaptation data. We have compared the basic fMLLR approach with proposed improvements from Section 3. Results of basic fMLLR, i-fMLLR, r-fMLLR with $Th_\gamma = 0.5$, and their combination ri-fMLLR, with increasing number of adaptation sentences, are depicted in Figure 2. According to Figure 2, i-fMLLR outperforms the basic fMLLR, but better results are obtained for r-fMLLR. Combination ri-fMLLR performs worst. Since the threshold prunes the amount of data used for accumulation of statistics the initialization outweighs. Hence, the adapted model is more tightened to

Table 1. Czech Telephone Corpus. Word Error Rate (WER)[%] of unadapted system and refined fMLLR adapted system.

	WER[%]
unadapted system	33.82
basic fMLLR	23.46
r-fMLLR ($Th_\gamma = 0.5$)	23.25
$Nbest$-fMLLR ($N = 1$)	23.23
$Nbest$-fMLLR ($N = 2$)	23.33

the unadapted model. However, to preserve the robustness of the system (in the sense of well-conditioned fMLLR transformation matrices) it is useful to retain the initialization. Note that the robustly adapted system ri-fMLLR performs still better than basic fMLLR.

In order to prove the validity of refinement parameters we have carried out experiments also on the SD-E set described in Section 4.2 and results can be found in Table 2. Results proved the generality of refinement values estimated on the CzT set.

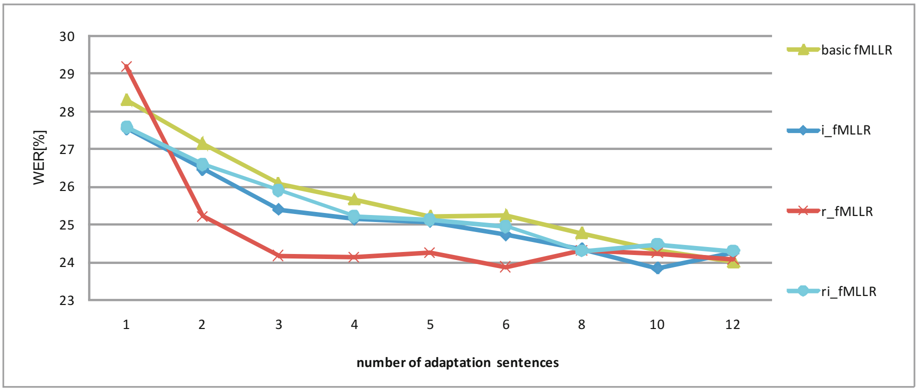


Fig. 2. Czech Telephone Corpus. Word Error Rate (WER)[%] of fMLLR adapted models for increasing number of adaptation sentences.

Table 2. SpeechDat-East Corpus. Word Error Rate (WER)[%] of unadapted system, refined fMLLR with $Th_\gamma = 0.5$ (r-fMLLR), initialized fMLLR (i-fMLLR) and combination of both (ri-fMLLR).

	WER[%]
unadapted system	55.18
basic fMLLR	46.12
r-fMLLR	44.50
i-fMLLR	44.66
ri-fMLLR	45.02

5 Conclusion

In this paper we have proposed procedures of robust estimation of adaptation parameters. We have investigated initialization of transformation matrices and refinement methods used to discard improper mixture components from the accumulation process of statistics. These techniques were applied to fMLLR approach, however they can be utilized also for any other estimations of linear transformations based on maximum likelihood. The main effort was dedicated to examination of the behavior of robustly adapted speech recognition system. Even if not all the proposed methods provided increase in systems performance, regarding the discussion in previous sections, the robustness of the system enhances. For example, such an approach is well suited for the task of adaptation in real-time recognition.

References

1. Psutka, J., Šmídl, L., Pražák, A.: Searching for a Robust MFCC-Based Parameterization for ASR Application. In: SIGMAP, Lisabon, pp. 196–199 (2007) ISBN: 978-989-8111-13-5
2. Pražák, A., Zajíc, Z., Machlica, L., Psutka, J.V.: Fast Speaker Adaptation in Automatic Online Subtitling. In: SIGMAP, Italy, pp. 126–130 (2009)
3. Gauvain, L., Lee, C.H.: Maximum A-Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. IEEE Transactions SAP 2, 291–298 (1994)
4. Gales, M.J.F.: Maximum Likelihood Linear Transformation for HMM-based Speech Recognition. Tech. Report, CUED/FINFENG/TR291, Cambridge Univ. (1997)
5. Povey, D., Saon, G.: Feature and Model Space Speaker Adaptation with Full Covariance Gaussians. In: Interspeech, paper 2050-Tue2BuP.14 (2006)
6. Gales, M. J. F.: The Generation and Use of Regression Class Trees for MLLR Adaptation. Cambridge University Engineering Department (1996)
7. Yu, K.: Adaptive Training for Large Vocabulary Continuous Speech Recognition. Ph.D. thesis, Hughes Hall College and Cambridge University Engineering Department (2006)
8. Li, Y., et al.: Incremental On-line Feature Space MLLR Adaptation for Telephony Speech Recognition. In: International Conference on Spoken Language Processing, Denver (2002)
9. Byrne, W., Gunawardana, A.: Discounted Likelihood Linear Regression for Rapid Adaptation. In: Eurospeech, Budapest, pp. 203–206 (1999)
10. Pollak, P., et al.: SpeechDat(E) – Eastern European Telephone Speech Databases, XLDB – Very Large Telephone Speech Databases. In: European Language Recources Association (ELRA), Paris (2000)
11. Young, S., et al.: The HTK Book (for HTK Version 3.4). Cambridge University Engineering Department (2001-2006)

A Priori and A Posteriori Machine Learning and Nonlinear Artificial Neural Networks

Jan Zelinka, Jan Romportl, and Luděk Müller

The Department of Cybernetics, University of West Bohemia, Czech Republic
SpeechTech s.r.o., Czech Republic
{zelinka@,rompi}@kky.zcu.cz, ludek.muller@speechtech.cz

Abstract. The main idea of a priori machine learning is to apply a machine learning method on a machine learning problem itself. We call it “a priori” because the processed data set does not originate from any measurement or other observation. Machine learning which deals with any observation is called “posterior”. The paper describes how posterior machine learning can be modified by a priori machine learning. A priori and posterior machine learning algorithms are proposed for artificial neural network training and are tested in the task of audio-visual phoneme classification.

1 Introduction

In this paper we are focusing on a function approximation problem, which is a branch of a supervised machine learning problem. In the function approximation problem the training set S consists of a finite number of pairs $(x; t)$ where x is an argument and t is a target. This problem defines then an approximating function $y = f_0(x, \theta_0)$ and a criterial function $\varepsilon_0(\theta_0, \mathfrak{S}_0)$, where θ_0 is the approximating function parameter and $\mathfrak{S}_0$ is the training set or predetermined statistics computed from the training set.

The goal of the problem is to search for the optimal parameter $\hat{\theta}_0$. The parameter $\hat{\theta}_0$ is optimal if $\varepsilon_0(\hat{\theta}_0, \mathfrak{S}_0) \leq \varepsilon_0(\theta_0, \mathfrak{S}_0)$. If the function approximation problem is unambiguous, the function $\hat{\theta}_0 = \hat{f}_1(\mathfrak{S}_0)$ exists in a mathematical point of view. To avoid nonconstructive reasoning we must consider only computable functions. Unfortunately, a function $\hat{f}_1$ is computable only in some very simple cases, such as least square error in linear regression. The simplest solution is thus to predetermine function $\theta_0 = f_1(\mathfrak{S}_0, \theta_1)$, where θ_1 is a parameter according to [1]. Therefore, the f_1 space dimension equals to the θ_1 space dimension. If $\mathfrak{S}_0$ is a vector of statistics, the simplest form of f_1 is the function $f_1(\mathfrak{S}_0, \theta_1) = \theta_1 \mathfrak{S}_0$. In this paper we have focused only on this form.

The aim of construction of a function approximation method is computation of the optimal parameter $\hat{\theta}_1 = \arg\max_{\theta_1} \varepsilon_1(\theta_1)$, where ε_1 is a criterial function. There are several meaningful criterial functions for θ_1 determination. The first criterial function is the mean of the “basic” criterial function ε_0, i.e. $\varepsilon_1(\theta_1) = \mathrm{E}\left\{\varepsilon_0(f_1(\mathfrak{S}_0, \theta_1))\right\}$. Another meaningful criterial function is

$$\varepsilon_1(\theta_1) = \mathrm{E}\left\{\|\theta_0 - f_1(\mathfrak{S}_0, \theta_1)\|^2\right\}.$$

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 472–479, 2010.

This choice leads to the optimal parameter

$$\hat{\theta}_1 = \mathrm{E}\left\{\theta_0 \mathfrak{S}_0^{\mathrm{T}}\right\} \left(\mathrm{E}\left\{\mathfrak{S}_0 \mathfrak{S}_0^{\mathrm{T}}\right\}\right)^{-1} .$$

Unfortunately, analytical solution of these means is impossible for common PDFs of $\mathfrak{S}_0$ and θ_1; moreover, usage of such PDFs, for which the means are analytically solvable, seems to be highly speculative. Therefore, a numerical method for θ_1 computing must be applied.

Although the function approximation problem differs from the problem of θ_1 estimation, one way of θ_1 computing is to consider it to be the function approximation problem – and our paper describes two numerical methods based on this methodological assumption.

However, these methods cannot do without relying on some analytically solved function approximation problem – the one we have made use of is a one-layer ANN with weights estimation algorithm which is optimal in compliance with MSE.

2 A Priori and A Posteriori Machine Learning

A priori and a posteriori machine learning basically differs in how the training set is acquired: the training set for the a priori machine learning is acquired without any real observation. The goal of the a priori machine learning is to construct a method for estimation of the parameter θ_0. Its training data consist of a finite number of pairs $(x;\, f_0(x, \theta_0))$, where x and θ_0 are randomly generated. The value of θ_0 is constant for each training sample. The training set thus consists of a finite number of pairs $(\mathfrak{S}_0(TD), \theta_0)$, where TD is the training sample obtained using θ_0. The optimal parameter of the approximating function $\theta_0 = f_1(\mathfrak{S}_0, \theta_1)$ is estimated by means of the function $\hat{\theta}_1 = \hat{f}_2(\mathfrak{S}_1)$, where $\mathfrak{S}_1$ is a vector of statistics computed from the training set.

There are two algorithms for estimation of θ_1. The first algorithm progresses in following steps:

1. For $i = 1, 2, \ldots, N$ do:
   (a) Generate randomly the parameter θ_0^i.
   (b) Generate randomly T examples of the input x into the set X.
   (c) Compute the set

$$TD^i = \left\{(x;t)\,;\, x \in X, t = f_0(x, \theta_0^i)\right\} .$$

2. Compute the set

$$S = \left\{\left(\mathfrak{S}_0(TD^i); \theta_0^i\right); i = 1, 2, \ldots, N\right\} .$$

3. The result of the algorithm is $\theta_1 = \hat{f}_2(\mathfrak{S}_1(S))$.

The constants N and T are fixed beforehand.

The second proposed algorithm is an iterative a posteriori modification of the first algorithm [2]. Here the training set S consists of pairs $(x; t)$, where x is an example of the input and t is its respective target. The algorithm progreses in following steps (the number of iterations is fixed to M):

1. Compute initialization θ_1^0 of the parameter θ_1 (e.g. as a result of the first algorithm).
2. For $i = 1, 2, \ldots, M$ do:
   (a) Estimate the parameter $\theta_0^i = f_1(\mathfrak{S}_0, \theta_1^{i-1})$.
   (b) For $j = 1, 2, \ldots, N$ do:
      i. Generate randomly the parameter $\theta_0^{i,j}$ from the close neighborhood of the parameter θ_0^i.
      ii. Select randomly T examples of the input x from the training data S into the set X.
      iii. Compute the set

$$TD^{i,j} = \left\{ (x;t) \, ; x \in X, t = f_0(x, \theta_0^{i,j}) \right\}.$$

   (c) Compute the set

$$S^i = \left\{ \left(\mathfrak{S}_0(TD^{i,j}); \theta_0^{i,j} \right); j = 1, 2, \ldots, N \right\}.$$

   (d) Compute $\theta_1^i = \hat{f}_2(\mathfrak{S}_1(S^i))$.
3. The result of the algorithm is θ_1^M.

The first algorithm offers a general method for function approximation, whereas the second algorithm is a method for function approximation specialized to the given training set.

3 Choice of Statistics

To avoid wholly heuristic choice of statistics, we introduced some more justified considerations. The fisrt consideration is about the optimal parameters estimation for function $y = A \cdot x$. If the number y is a posterior estimation, the special criterial function can be used: $\varepsilon(A) = \sum_i (2t_i - 1)(2y_i - 1)$. The criterial function prefers bigger negative values of y for $t = 0$ and bigger positive values for $t = 1$. Although the criterial function has neither global nor local maximum, a gradient algorithm can be applied. The gradient is $\frac{\partial \varepsilon}{\partial A} = 4 \sum_i t_i x_i - 2 \sum_i x_i$. The gradient does not depend on the parameter A, thus all steps of the gradient algorithm are possible at once. Hence, the statistics $\sum_i t_i x_i$ and $\sum_i x_i$ are useful for posterior estimation or logical function approximation. Moreover, the statistics $\sum_i x_i$ can be ignored when the mean subtraction is applied.

The second consideration is a more general consideration about data sets. Alternative representation of the data set $S = \{x; x \subset \Re\}$ is a sequence of statistics $(\mathfrak{S}_n(S))_{n=1}^{\infty}$. There is a trivial proof of the theorem that there exists a statistics $\mathfrak{S}_n$ such that the projection of the finite set $S \neq \emptyset$ on the sequence $(\mathfrak{S}_n(S))_{n=1}^{\infty}$ is isomorphism. Nontrivial statistics which privide isomorphic projection are sample moments $\mathfrak{S}_n(S) = \frac{\sum_{x \in S} x^n}{\|S\|}$; analogically for $S = \{x \in \Re^N\}$ where $N > 1$. Consequently a shortened sequence $(\mathfrak{S}_n(S))_{n=1}^{m}$ can be seen as an approximation of a set S.

4 Artificial Neural Networks

Literature describes at least applications of the first algorithm in the area of Artificial Neural Networks (ANN): for example in [3,4] an ANN is used for parameter estimation and it is trained a priori. However, this ANN does not estimate parameters of another ANN – and this is the task we would like to use it in.

All ANNs we use have only one layer. The i-th output of an ANN with one layer is given by the formula

$$y_i = f(x,\theta) = \varphi\left(\sum_{j=1}^{n_x} w_{i,j}x_j + b_i\right), \tag{1}$$

where $i = 1, 2, \dots, n_y$, $w_{i,j} \in \Re$, $b_i \in \Re$ and $\theta = \left(w_{1,1}, \dots, w_{n_y,n_x}, b_1, \dots, b_{n_y}\right)$. If $\varphi(\xi) = \xi$ and the criterion function $\varepsilon(\theta, S)$ is MSE, i.e.

$$\varepsilon(\theta, S) = \sum_{i=1}^{n_S} \|t_i - f(x_i,\theta)\|^2, \tag{2}$$

where $S = \{(x_i; t_i); i = 1, 2, \dots, n_S\}$ is the training set, then the optimal parameter $\hat{\theta}$ can be computed analytically as the solution of the set of linear equations. For such computation only these statistics are necessary:

$$\mathfrak{S}_A(S) = \frac{1}{n_S}\sum_{i=1}^{n_S}\begin{bmatrix} x_i \\ 1 \end{bmatrix}\left[x_i^{\mathrm{T}}; 1\right], \mathfrak{S}_B(S) = \frac{1}{n_S}\sum_{i=1}^{n_S} t_i\left[x_i^{\mathrm{T}}; 1\right]. \tag{3}$$

Using these statistics the optimal parameters could be computed using the formula

$$\begin{pmatrix} w_{1,1} & \cdots & w_{1,n_x} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ w_{n_y,1} & \cdots & w_{n_y,n_x} & b_{n_y} \end{pmatrix} = \mathfrak{S}_B \mathfrak{S}_A^{-1}. \tag{4}$$

It is possible to use pseudoinversion instead of inversion if the matrix $\mathfrak{S}_A$ is singular.

We have used this ANN as the function f_1 and both proposed algorithms estimate parameters of this ANN and we have used the statistics

$$\mathfrak{S}_0 = \frac{1}{n_S}\sum_{i=1}^{n_S} t_i x_i^{\mathrm{T}}. \tag{5}$$

Figures 1 and 2 schematically illustrate both algorithms.

In addition to the described ANN we have also used one simple artificial neuron given by the formula

$$y = \sigma\left(wT(x) + b\right), \tag{6}$$

where $x = (x_1, \dots, x_n)^{\mathrm{T}}$ is an input vector and y is a neuron's output, $T(x) = (x_1, \dots, x_n, x_1x_1, \dots, x_1x_n, \dots, x_nx_n)^{\mathrm{T}}$ and the vector w and the number b are

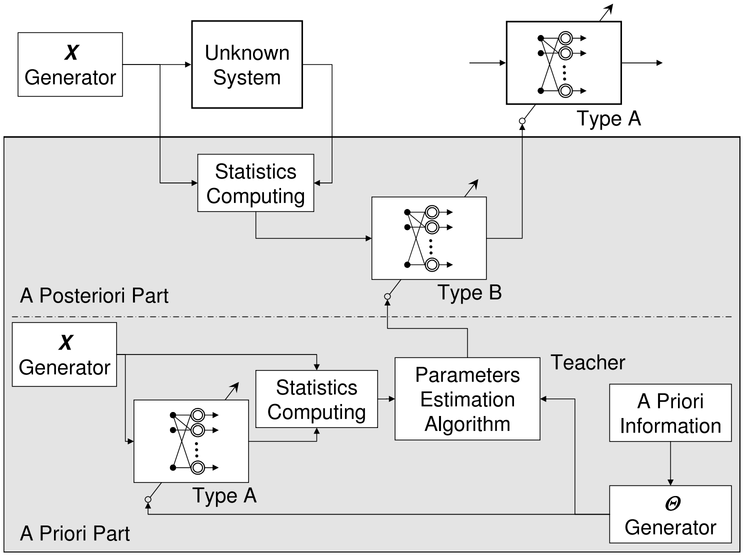


Fig. 1. The first algorithm schematization

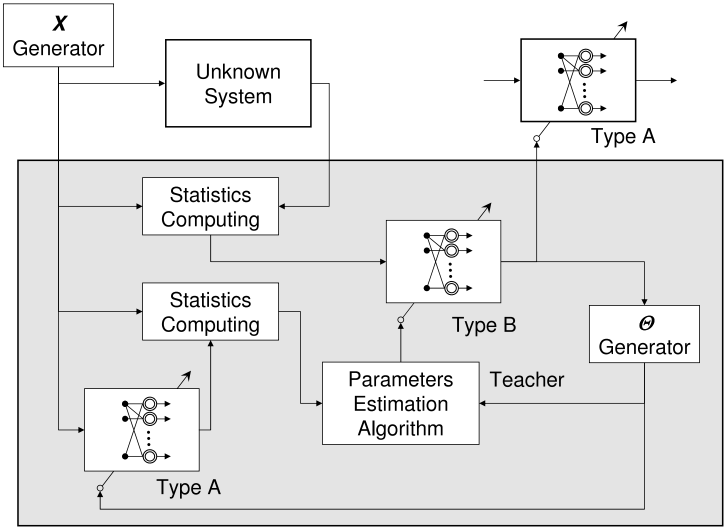


Fig. 2. The second algorithm schematization

parameters of the neuron. We have used the statistics $\mathfrak{S}_0 = (\mathfrak{S}_u(S), \mathfrak{S}_v(S), \mathfrak{S}_w(S))$, where

$$\mathfrak{S}_u(S) = \frac{1}{n_S}\sum_{i=1}^{n_S} t_i T(x_i)^{\mathrm{T}}, \quad \mathfrak{S}_v(S) = \frac{1}{n_S}\sum_{i=1}^{n_S} T(x_i), \mathfrak{S}_w(S) \quad = \frac{1}{n_S}\sum_{i=1}^{n_S} t_i .$$

5 Experiments and Results

We have tested our approach in two different experiments. In the first experiment we have used a priori machine learning (i.e. the first proposed algorithm) for parameter estimation of the artificial neuron described in (6). The algorithm has reached 81,000 iterations during the process. The results for 16 commonly used training sets (based on binary logical operators) is shown in Table 1.

Table 1. The first experiment – a priori learning with 16 training sets based on binary logical operators

i	x_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i	t_i
1	0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
3	1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
4	1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
i	x_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i	y_i
1	0 0	0.01	0.01	0.05	0.07	0.05	0.06	0.24	0.28	0.72	0.76	0.94	0.95	0.93	0.95	0.99	0.99
2	0 1	0.01	0.05	0.01	0.07	0.71	0.94	0.76	0.95	0.05	0.24	0.06	0.29	0.93	0.99	0.95	0.99
3	1 0	0.01	0.05	0.72	0.94	0.01	0.07	0.76	0.95	0.05	0.24	0.93	0.99	0.06	0.28	0.94	0.99
4	1 1	0.01	0.71	0.05	0.94	0.05	0.93	0.24	0.99	0.01	0.76	0.07	0.95	0.06	0.95	0.29	0.99

In the second experiment we have used the audio-visual Czech speech database described in [5], where a multi-layer ANN estimates posteriors from acoustic and from visual modality separately (see for example [6]). The goal of our second experiment is to create fusion of these posteriors to achieve higher accuracy of the audio-visual speech data classification into phonemes – this is performed by the aforementioned single-layer ANN. For this experiment the acoustic modality was noised with white noise; the *SNR* was 0.

We have decided to evaluate the posteriors estimation quality as the classification accuracy: an observation o is classified using the mean of posteriors estimation $\tilde{p}(\omega|o)$ as $\arg\max_{\omega\in\Omega} \tilde{p}(\omega|o)$, where Ω is a phonetic alphabet (our phonetic alphabet consists of $n_\Omega = 55$ phonemes).

The input vector of the single-layer ANN is

$$x = \left(p_1^A, \dots, p_{n_\Omega}^A, p_1^V, \dots, p_{n_\Omega}^V, p_1^A p_1^V, \dots, p_{n_\Omega}^A p_{n_\Omega}^V, \right), \tag{7}$$

where p_i^A is the i-th posterior estimation from the acoustic modality and p_i^V is the i-th posterior estimation from the visual modality.

We have selected two disjoint data sets from the corpus: Φ and Ψ. Each data set contains 100,000 examples. The set Φ has been used for training and the set Ψ for testing.

Since we have used ANN also as a training method, there is a danger that not only the resulting fusion method but also the parameter estimation method can become overtrained. High accuracy for the training set and low accuracy for the testing set indicates the fusion method is overtrained; high accuracy in case the fusion method is trained from the training set which is exactly the same as the one used for the parameter estimation training, and low accuracy in other cases indicates the parameters estimation method is overtrained. Both types of overtraining can occur concurrently. To indicate these phenomena we have tested all four combinations of using the training and testing sets for the fusion method training, but only the set Φ was used for the parameter estimation training.

In addition to both proposed algorithms for ANN parameter estimation, several common posteriors fusion methods have been tested: Average of Posteriors (AP), Multiple of Posteriors (MP) and Entropy Based Fusion (EBF) [7]. These common methods do not need any training set, thus their accuracies do not depend on it. All results are shown in Table 2.

Table 2. The second experiment – fusion of posteriors in audio-visual phoneme classification

Training Set	Φ	Ψ	Φ	Ψ
Testing Set	Φ	Φ	Ψ	Ψ
$\left(p_1^V, \ldots, p_{n_\Omega}^V\right)^\mathrm{T}$	38.4%	38.4%	37.3%	37.3%
$\left(p_1^A, \ldots, p_{n_\Omega}^A\right)^\mathrm{T}$	62.0%	62.0%	60.7%	60.7%
AP	63.2%	63.2%	62.2%	62.2%
MP	65.7%	65.7%	65.3%	65.3%
EBF	63.1%	63.1%	62.0%	62.0%
ANN (MSE)	66.3%	66.0%	64.8%	66.3%
ANN (1. alg.)	58.7%	57.9%	58.0%	58.4%
ANN (2. alg., 1. it.)	62.2%	62.2%	60.8%	61.4%
ANN (2. alg., 26. it.)	67.1%	66.6%	65.5%	66.2%

6 Conclusion and Future Work

The main idea of this paper can be simply expressed by the name of the Burnett's article "Learning to learn in a virtual world" [8]. The described a priori machine learning algorithm is only the simplest realization of the sketched idea, i.e. the a priori modification of metalearning [9] – further work will be focused on more elaborate algorithms. Our approach keeps away the self-reference problem [10] because the algorithms are not so strong to provide any self-reference possibility. In the same time our experiments have proved that our approach is fully efficient: the resulting posteriors combination method is – in the worst tested case – as accurate as the described standard ANN and other common methods. Moreover, the resulting ANN parameter estimation algorithm is significantly simpler.

Acknowledgements

This research was supported by the Ministry of Education of the Czech Republic, project No. 2C06020, and by the Grant Agency of the Czech Republic, project No. GAČR 102/08/0707.

References

1. Heskes, T.: Empirical Bayes for Learning to Learn. In: Proceedings of ICML, pp. 367–374. Morgan Kaufmann, San Francisco (2000)
2. Vilalta, R., Drissi, Y.: A Perspective View and Survey of Meta-Learning. Artificial Intelligence Review 18, 77–95 (2002)
3. Kumar, R.: A Neural Network Approach to Rotorcraft Parameters Estimation (2007)
4. Hering, P., Šimandl, M.: Gaussian Sum Approach with Optimal Experiment Design for Neural Network, Honolulu, pp. 425–430. ACTA Press (2007)
5. Císař, P., Železný, M., Krňoul, Z., Kanis, J., Zelinka, J., Müller, L.: Design and Recording of Czech Speech Corpus for Audio-Visual Continuous Speech Recognition. In: AVSP 2005, Vancouver Island, pp. 1–4 (2005)
6. Potamianos, G., Neti, C., Iyengar, G., Helmuth, E.: Large-Vocabulary Audiovisual Speech Recognition: A Summary. In: Proc. Works. Signal Processing, Johns Hopkins Summer 2000 Workshop, pp. 619–624 (2001)
7. Grézl, F.: TRAP-Based Probabilistic Features for Automatic Speech Recognition. Ph.D. thesis, MUNI (2007)
8. Burnett, R.: Learning to Learn in a Virtual World, Milan, Italy. AERA (1999)
9. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. Springer Publishing Company, Heidelberg (2008) (incorporated)
10. Schmidhuber, J.: Steps Towards 'Self-Referential' Neural Learning: A Thought Experiment. Technical Report CU-CS-627-92, Department of Computer Science and Institute of Cognitive Science, University of Colorado, Boulder, Boulder, CO (1992)

Posterior Estimates and Transforms for Speech Recognition

Jan Zelinka, Luboš Šmídl, Jan Trmal, and Luděk Müller

The Department of Cybernetics, University of West Bohemia, Czech Republic
{zelinka,smidl,jtrmal,muller}@kky.zcu.cz

Abstract. This paper describes ANN based posterior estimates and their application to speech recognition. We replaced the standard back-propagation with the L-BFGS quasi-Newton method. We have focused only on posterior based feature vector extraction. Our goal was a feature vector dimension reduction. Thus we designed three posterior transforms to space with dimensionality 1 or 2. The designed transforms were tested on the SpeechDat-East corpus. We also applied the introduced method on a Czech audio-visual corpus. In both cases the methods leads to significant word error rate decrease.

Keywords: artificial neural networks, posteriors, speech recognition.

1 Introduction

In the paper we described posterior estimates and posterior transformates and their application in speech recognition. The posteriors are conditional probabilities $p(u|o)$ of some speech unit u given the occurrence of some feature vector o. In the paper a unit is a phone or state of a monophone. Application of Artificial Neural Networks (ANN) is wide-ranging [1,2] and ANNs also play a major role in the posterior estimates problem.

In audio-visual speech recognition when only a feature fusion is performed both posterior estimates are combined into one single vector of posterior. The combination that is applied in our experiments is not only common simple combination such as entropy based combination or geometric fusion but it is a combination of several common combinations.

There are two basic kinds of posterior estimates applications. The first kind of application is an HMM hybrid where posterior estimates serve as surrogate labeler. The second kind of application is a regular HMM where posterior estimates serve as a part of feature vector. In this paper we focused only on the second kind. We transformed vector of posterior to a vector with one or two features. To the small vector the standard PLP parameterization is added. The schemes in both (audio and audio-visual) feature vector computation are shown in Figure 1.

2 Posterior Estimates

The standard and successful approach to posterior estimate is the use of ANN. In this paper we focused only on this approach. We used the TRAPS [3,4] method as an acoustic signal parameterization method for posterior estimate.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 480–487, 2010.

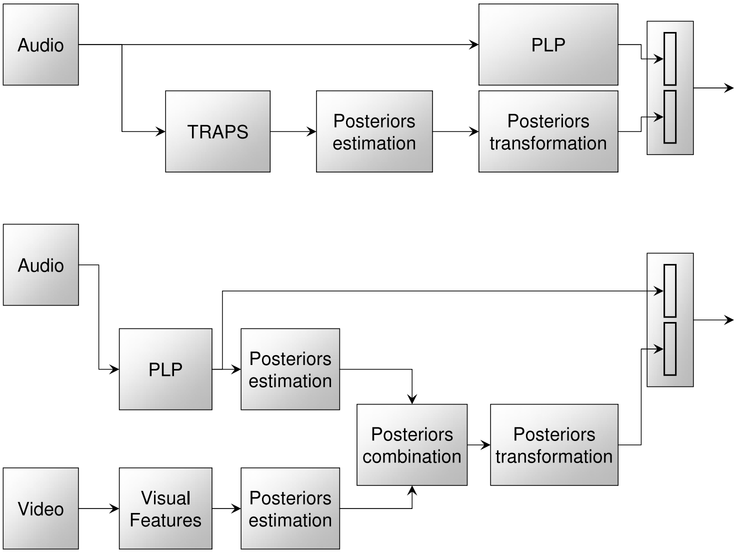


Fig. 1. Two schemata of feature vector computing

There are two problems when an ANN is used. The first problem is the choice of the ANN structure. We avoid the first problem by virtue of using the structure as a structure described in literature (see Figure 2). The chosen ANN structure has two layers. Activation functions of neurons in a hidden layer are sigmoid functions and activations function in output layer are softmax:

$$y_i = \frac{e^{\xi_i}}{\sum_j e^{\xi_j}}, \tag{1}$$

where y_i is an output of an i-th output neuron and ξ_i is an input of i-th activation function. The TRAPS signal parameterization method produces a feature vector with dimension equal to 330. The highest number of neurons in the hidden layer is 1500. The highest number of estimated posteriors is equal to 111.

The second and more serious problem is the choice of training algorithm. To train an ANN the researches usually use some variant of backpropagation with dynamic change of learning constant – the Bold Driver method or similar approaches are common [5]. From our experience, these methods show poor convergence rate when used to train ANNs with many parameters (1M and more). The slow convergence results in an increased number of gradient evaluation, which is the most time consuming procedure during the training (given the fact that the size of speech corpora is usually very large – hundred(s) of hours).

The backpropagation algorithm as a special case of gradient algorithm can be interpreted as local criterion function approximation which is approximated by a linear hyper-plane. The momentum method is a similar approximation. A Newton method approximates the criterion function by a quadratic hyper-plane. Thus at least a local extreme can be estimated more accurately and consequently the iteration process can be accelerated.

But the enormous number of parameters of the used multi-layer ANN structure precludes using the Newton method regardless of the possible instability of it. Therefore, application of a pseudo-Newton method approximating the Hessian with the gradient is necessary. We tested the L-BFGS (Limited memory Broyden-Fletcher-Goldfarb-Shanno [6]) method[1] and the iRPROP$^+$ [7] method and both performed significantly faster than the backpropagation with dynamic learning constant that we had used before. Moreover, for both of these methods it is much simpler to "guess" the initial settings. However, the evaluation of improvements or suitability of these methods is out of scope of this paper.

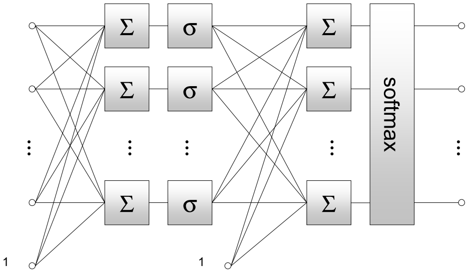


Fig. 2. Schema of used ANN

The used iterative algorithm is stochastic because of the random initialization. All the other steps are deterministic. In an alternative algorithm a stochastic version was used instead of each deterministic step s.

$$\theta_{t+1} = s\left(\theta_t\right) + \gamma_t e_t, \tag{2}$$

where e_t is a random vector (with standard normal distribution) and γ_t is given by the formula

$$\gamma_t = \begin{cases} \gamma_0 & t = 1 \\ \gamma_\uparrow \gamma_{t-1} & \varepsilon(\theta_{t-1}) = \varepsilon(\theta_{t-2}) \\ \gamma_\downarrow \gamma_{t-1} & \varepsilon(\theta_{t-1}) \neq \varepsilon(\theta_{t-2}) \end{cases}, \tag{3}$$

where $\gamma_\uparrow > 1$ and $0 < \gamma_\downarrow < 1$ are chosen constants and ε is the monitored criterion, which is in general not identical with minimizing criterion. This modification is a compensation of the inferior initialization (i.e. θ_0).

[1] L-BFGS toolbox we used can be found at http://www.chokkan.org/software/liblbfgs/

3 Posterior Transforms

The pure and simple aim of posterior transform is a decrease in word error rate. There are usual posterior transforms such as logarithm and linear transform. Parameters of the linear transform are computed by means of PCA, LDA or another similar method. Unfortunately, this posterior transforms lead to a relatively high feature space dimension. In this paper we have tried to find a transform which not only decreases a word error rate but which even leads to very small feature vector dimension.

As Cantor proved, there are bijective functions $\Re^n \mapsto \Re$ for all $0 < n \in N$, which is of course nonlinear. Beside this there is a injective linear function from a finite set $A \subset \Re^m$ to a set $B \subset \Re^n$ if the cardinality of the set A is less than or equal to the cardinality of the set B and if all elements of A are linearly independent. Thus the maximal number of elements of set A is m. In case of posterior estimations space $\langle 0; 1\rangle^{n_p}$ we have chosen the space quantization $A = \left\{(\delta_{i,1}, \delta_{i,2}, \dots, \delta_{i,n_p})|i = 1, 2, \dots, n_p\right\}$ where $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ otherwise. The applied quantization projection $\langle 0; 1\rangle^{n_p} \mapsto A$ is given by the formula:

$$a_i = \delta_{i,\arg\max_j p_j} \tag{4}$$

where $(p_1, p_2, \dots, p_{n_p}) \in \langle 0; 1\rangle^{n_p}$ and $(a_1, a_2, \dots, a_{n_p}) \in A$. This is a simple and natural choice. But there is an inexhaustible number of choices of the set B.

With no additional information (about posterior estimates or about following classification) there are only distance base criterions. We have chosen so called Discriminative Ration (DR)

$$DR = \frac{d}{D}, \tag{5}$$

where

$$d = \min_{b_1, b_2 \in B, b_1 \neq b_2} \|b_1 - b_2\|, \; D = \max_{b_1, b_2 \in B, b_1 \neq b_2} \|b_1 - b_2\|. \tag{6}$$

The maximum of the criterion is desired.

In case of $n = 1$, i.e. $B \subset \Re$, it is easy to prove that the optimal set B is a set

$$B = \left\{a \cdot i + b | i = 0, 1, \dots, n_p - 1\right\}, \tag{7}$$

where $a \neq 0, b \in \Re$. Hence $DR_1 = \frac{1}{n_p - 1}$. We fixed $a = \frac{1}{n_p - 1}$ and $b = 0$.

Unfortunately, in case of $n = 2$ there is no optimal solution for some n_p. Therefore we heuristically proposed two different suboptimal solutions. The first set $B \subset \Re^2$ is the set

$$B = \left\{\left[\sin\left(\frac{2\pi i}{n_p}\right); \cos\left(\frac{2\pi i}{n_p}\right)\right]^{\mathrm{T}} | i = 0, 1, \dots, n_p - 1\right\}. \tag{8}$$

For this set $\sqrt{\frac{1 - \cos\left(\frac{2\pi}{n_p}\right)}{\frac{3}{2}}} \geq DR_2 \geq \sqrt{\frac{1 - \cos\left(\frac{2\pi}{n_p}\right)}{2}}$. The second set is a set of approximately square grid of points. In this case $DR_3 = \frac{1}{\sqrt{2\overline{n}_p}}$, where $\overline{a}$ is the minimal square number which is bigger than or equal to $a \in N$ ($\overline{111} = 121$). Very small dimensionality increasing is the motivation for design this transforms.

Despite the fact that the sets A and B are determined, the projection $A \mapsto B$ is not unambiguous. There are $n_p!$ different injective functions ($111! \cong 1.76 \cdot 10^{180}$), thus all possible projections definitely could not be investigated. It is possible to propose some heuristic projection choice such as a choice where the classes confused more often are geometrically closer than the other ones but we have decided to choose random projection (e.g. sequence of points is given by the lexicographical order of units).

4 Experiments and Results

In our experiments two different corpora were used in our tests. The first corpus was SpeechDat-East (SD-E) [8]. SD-E contains telephone speech in 5 languages Czech, Polish, Slovak, Hungarian, and Russian. Only the Czech part of SD-E was used. The acoustic HMM was trained on 700 speakers with 50 utterances for each speaker (cca 5 sec. per sentence). For testing purposes 150 speakers were chosen with 50 utterances for each speaker. The vocabulary consisted of 7,737 words. No OOV words were present. The audio modality was recorded with 8 kHz sampling frequency. Triphones were model using 3 states HMM with 8 GMM in each state. For the recognition a zerogram language model was applied. Each recording was segmented into segments corresponding to states of monophones. The number of posteriors was 111. The number of neurons in hidden layer was 1,500. The results for WER are shown in Table 1.

Table 1. Contributions of the proposed posterior transforms (SD-E)

Parameterization	Dimension	Corr (sentence)	Corr (word)	Acc (word)
PLP + Δ + ΔΔ	36	33.08%	71.26%	63.02%
PLP + Δ + ΔΔ + 1D	37	41.21%	72.76%	71.35%
PLP + Δ + ΔΔ + 2D(circle)	38	41.98%	74.08%	70.95%
PLP + Δ + ΔΔ + 2D(square)	38	40.03%	74.58%	70.39%

As the second corpus we used the Czech audio-visual corpus (CAVC) [9]. The audio was recorded with 44.1 kHz sampling frequency. 150 sentences were reserved for training and 50 sentences were reserved for testing. The vocabulary consisted of 345 words. No OOV words were present. Monophones were modeled using 3 states HMM with 10 GMM in each of the states. For the recognition, a zerogram language model was applied. Each recording was segmented into segments corresponding to monophones. The number of posteriors was 55.

Table 2. Backpropagation compared with L-BFGS (CAVC)

method	n_{hidd}	$SNR = +\infty$	$SNR = 10$	$SNR = 0$
backpropagation	100	$Acc = 72.12\%$	$Acc = 65.67\%$	$Acc = 56.43\%$
backpropagation	300	$Acc = 73.87\%$	$Acc = 66.50\%$	$Acc = 54.90\%$
L-BFGS	100	$Acc = 73.71\%$	$Acc = 68.43\%$	$Acc = 60.63\%$

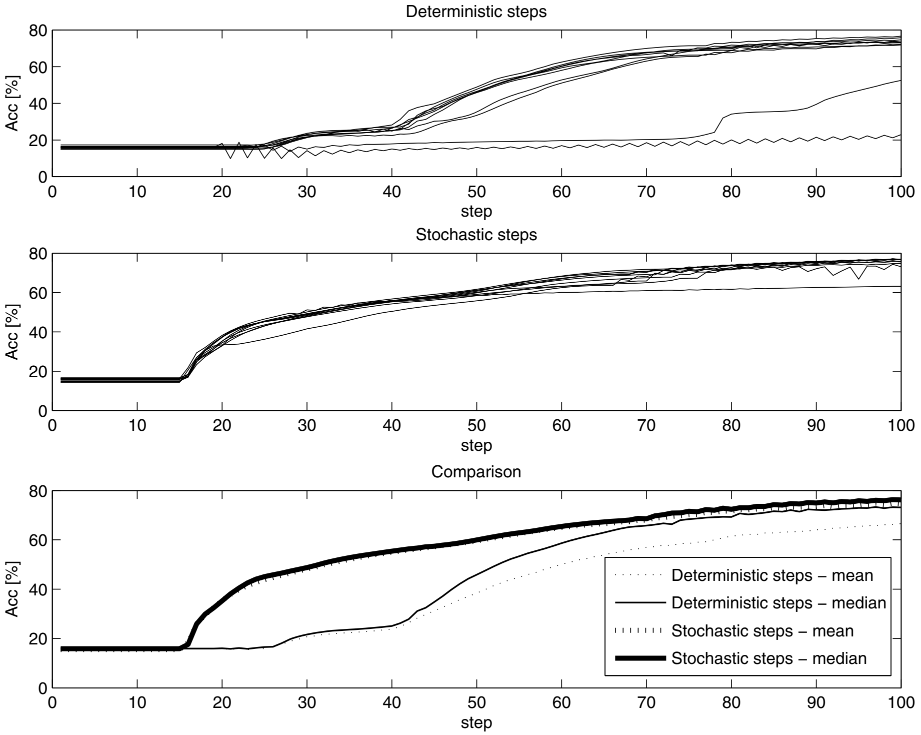


Fig. 3. The comparison between the deterministic and the stochastic steps (CAVC)

In the next experiment the contribution of the stochastic steps was considered. In the experiment, 10 training process were performed from 10 different initial conditions. Figure 3 shows the results for CAVC. The results indicate that stochastic steps lead to quicker convergence and it makes a training process more robust. In the second experiment with the corpus the contribution of L-BFGS algorithm was evaluated. All steps of all training algorithms were stochastic. The results are shown in Table 2 where n_{hidd} means number of neurons in a hidden layer. Only the acoustic modality was used and the criterion of posterior estimate quality is classification accuracy where the classification is the index into a posterior vector with the highest value.

The aim of the last experiment was to evaluate the contribution of the proposed transforms on CAVC. The correctness and accuracy of speech recognition were measured. Acoustic and video modalities were used in this experiment because used posterior estimates are result of combination of posterior estimates computed from both modalities. We used Bootstrapping method to estimate the confidence intervals for 95% and 99% [10]. The results are shown in Table 3.

Table 3. Contributions of proposed posterior transforms (CAVC)

Parameterization	$SNR = +\infty$	$SNR = 10$	$SNR = 0$
PLP + Δ + ΔΔ	**Corr=82.97,** Corr$_{95}$=82.35–83.59, Corr$_{99}$=82.15–83.78, **Acc=81.73,** Acc$_{95}$=81.09–82.38, Acc$_{99}$=80.88–82.57	**Corr=75.54,** Corr$_{95}$=74.86–76.23, Corr$_{99}$=74.63–76.46, **Acc=73.86,** Acc$_{95}$=73.14–74.58, Acc$_{99}$=72.91–74.82	**Corr=54.90,** Corr$_{95}$=54.08–55.73, Corr$_{99}$=53.83–56.01, **Acc=52.18,** Acc$_{95}$=51.36–53.01, Acc$_{99}$=51.10–53.30
PLP + 1D	**Corr=86.62,** Corr$_{95}$=86.10–87.12, Corr$_{99}$=85.94–87.28, **Acc=84.50,** Acc$_{95}$=83.93–85.06, Acc$_{99}$=83.77–85.22	**Corr=83.04,** Corr$_{95}$=82.46–83.61, Corr$_{99}$=82.29–83.79, **Acc=80.58,** Acc$_{95}$=79.94–81.20, Acc$_{99}$=79.75–81.42	**Corr=75.22,** Corr$_{95}$=74.53–75.89, Corr$_{99}$=74.32–76.08, **Acc=72.77,** Acc$_{95}$=72.05–73.47, Acc$_{99}$=71.83–73.68
PLP + 2D(circle)	**Corr=86.97,** Corr$_{95}$=86.46–87.47, Corr$_{99}$=86.30–87.63, **Acc=83.96,** Acc$_{95}$=83.37–84.54, Acc$_{99}$=83.18–84.71	**Corr=83.18,** Corr$_{95}$=82.62–83.74, Corr$_{99}$=82.45–83.89, **Acc=79.81,** Acc$_{95}$=79.17–80.44, Acc$_{99}$=78.98–80.62	**Corr=75.55,** Corr$_{95}$=74.87–76.21, Corr$_{99}$=74.67–76.42, **Acc=72.37,** Acc$_{95}$=71.64–73.09, Acc$_{99}$=71.42–73.33

5 Conclusion and Future Work

In this paper we have presented several methods we developed to improve accuracy of speech recognition using posterior estimate. In our experiments the contributions of all three proposed posterior transformates was demonstrated. Accuracy increments are considerable especially in audio-visual recognition for $SNR > -\infty$. Contribution of L-BFGS algorithm and its stochastic modification for posterior estimates in speech recognition was substantial too. In the future, we plan to use a similar method for posterior estimates and transformates in large vocabulary speech recognition system [11,12] for Czech and English language.

Acknowledgements

This research was supported by the Ministry of Education of the Czech Republic, project No. MŠMT LC536, by the grant of The University of West Bohemia, project No. SGS-2010-054, and by the Ministry of Education of the Czech Republic, project No. MŠMT LC536. The access to the METACentrum supercomputing facilities provided under the research intent MSM6383917201 is highly acknowledged.

References

1. Hering, P., Šimandl, M.: Gaussian Sum Approach with Optimal Experiment Design for Neural Network. In: Proceedings of the Ninth IASTED International Conference on Signal and Image Processing, Honolulu, pp. 425–430. ACTA Press (2007)

2. Šimandl, M., Hering, P.: Recursive Parameters Estimation and Structure Adaptation of Neural Network. In: Proceedings of the Eighth IASTED International Conference on Intelligent Systems and Control, Anaheim, pp. 78–83. ACTA Press (2005)
3. Schwarz, P., Matějka, P., Černocký, J.: Towards Lower Error Rates in Phoneme Recognition. LNCS (LNAI), pp. 465–472. Springer, Heidelberg (2004)
4. Schwarz, P., Matějka, P., Černocký, J.: Recognition of Phoneme Strings using Trap Technique. In: Proceedings of 8th International Conference Eurospeech, International Speech Communication Association, pp. 1–4 (2003)
5. Salomon, R., Hemmen, J.L.V.: Accelerating Backpropagation Through Dynamic Self-Adaptation. Neural Networks 9, 589–601 (1996)
6. Nocedal, J.: Updating Quasi-Newton Matrices with Limited Storage. Mathematics of Computation 35, 773–782 (1980)
7. Igel, C., Hüsken, M.: Empirical Evaluation of the Improved Rprop Learning Algorithms. Neurocomputing 50, 105–123,19 (2003)
8. Pollak, P.: Speechdat(e) – Eastern European Telephone Speech Databases. In: Proceedings LREC 2000 Satelite Workshop XLDB, Athens, Greece, pp. 20–25 (2000)
9. Císař, P., Železný, M., Krňoul, Z., Kanis, J., Zelinka, J., Müller, L.: Design and Recording of Czech Speech Corpus for Audio-Visual Continuous Speech Recognition. In: Proceedings of the Auditory-Visual Speech Processing International Conference 2005, AVSP 2005, Vancouver Island (2005)
10. Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: IJCAI, pp. 1137–1143. Morgan Kaufmann, San Francisco (1995)
11. Ircing, P., Psutka, J.V., Psutka, J.: Using Morphological Information for Robust Language Modeling in Czech ASR System. IEEE Transactions on Audio, Speech, and Language Processing 17, 840–847 (2009)
12. Prazák, A., Zajíc, Z., Machlica, L., Psutka, J.V.: Fast Speaker Adaptation in Automatic Online Subtitling. In: SIGMAP, pp. 126–130 (2009)

Part IV

Dialogue

"**Dialogue:** a discussion between two or more people or groups, especially one directed towards exploration of a particular subject or resolution of a problem: *interfaith dialogue*."
NODE (The New Oxford Dictionary of English), Oxford, OUP, 1998, page 509.

A Multimodal Dialogue System for an Ambient Intelligent Application in Home Environments

Nieves Ábalos*, Gonzalo Espejo, Ramón López-Cózar,
Zoraida Callejas, and David Griol

Dept. of LSI, CITIC-UGR, University of Granada, Spain
Dept. of Computer Science, Carlos III University of Madrid, Spain
{nayade,gonzaep}@correo.ugr.es, {rlopezc,zoraida}@ugr.es,
dgriol@inf.uc3m.es

Abstract. This paper presents a multimodal dialogue system called *Mayordomo* which aims at easing the interaction with home appliances using speech and a graphical interface within an Ambient Intelligence environment. We present the methods employed for implementing the system describing the design of the user-system interactions as well as additional features such as the management of user profiles to restrict the access to domestic appliances and customize the recognition grammars and the generated responses.

Keywords: spoken dialogue systems, multimodal systems, ambient intelligence, automatic speech recognition, spoken language understanding, dialogue management, text-to-speech synthesis.

1 Introduction

Ambient Intelligence (AmI) is a research area that has attracted a lot of efforts by the scientific community in the last years [1,2,3,4,5]. The aim of AmI is to create environments in which users are able to interact in a natural and transparent way with systems that help them carrying out their daily leisure and work activities.

As envisaged by the European Information Society Technology Advisory Group [6], AmI provides a vision of the Information Society where the emphasis is on a greater user friendliness without a detriment in service efficiency. Within this vision, people are surrounded by intelligent and intuitive interfaces that are embedded in all kinds of objects. That is, they are immersed in an environment that is capable of recognising and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way.

An AmI system uses contextual information that can be generated either by the environment or the user. This information can be employed to adapt the system to user preferences and needs, for example, switching on a light in the user's house [7,1]. In addition, some AmI systems use this contextual information to act proactively. For instance, the system can be able to play music while the user is coming into a room without requiring an explicit order.

* This research has been funded by the Spanish project HADA TIN2007-64718.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 491–498, 2010.

The rest of the paper is organized as follows. Section 2 introduces "*Mayordomo*", the multimodal dialogue system which we are developing. The section describes the characteristics of the system's interaction with different users and with the environment. Section 3 presents the conclusions and outlines possibilities for future work.

2 *Mayordomo* Dialogue System

Mayordomo is a multimodal dialogue system under development in our laboratory, which aims to centralize control of appliances in a home. The system is multimodal as it provides several methods for interaction with appliances. Specifically, users can employ either spontaneous speech or a traditional GUI interface based on keyboard and mouse.

The system has been designed to operate in an AmI environment in order to ease the interaction of users with such environment. For example, *Mayordomo* can find out the room in which the user is at any time. This localization information can be used to optimize the dialogue with the user, thus ridding him off about providing unnecessary information.

The system allows parental control of some appliances in order to restrict the interaction with them. For instance, parents can forbid that children watch TV after 10 p.m. The system administrator has privileges to perform special actions, for example, installing and uninstalling appliances and handling the parental control. The system creates a log of all actions carried out within the environment by any user.

2.1 Oral Interaction

We use Windows Vista Speech Recognition (WVSR) to implement the oral interaction. This package includes both the engine for automatic speech recognition (ASR) and the engine for text-to-speech synthesis (TTS). Windows Vista includes two development tools for programmers: SAPI 5.3 (Speech API) and System.Speech (.NET Framework 3.0 namespace). To implement the system we use System.Speech as it is oriented mainly to programming languages for Microsoft .NET. The package provides a collection of classes that enables ASR (System.Speech.Recognition classes) and TTS (System.Speech.Synthesis classes).

Automatic Speech Recognition (ASR). As mentioned above, each appliance has associated a configuration file that allows the user to control it orally. This is so because this file contains a specific grammar for interacting with the appliance that is used for ASR. This grammar is specified in SRGS (Speech Recognition Grammar Specification) format, which defines the syntax of grammars for ASR.

When specifying grammars for the different home appliances of a home, we can consider three strategies: (1) allowing keyword recognition using the specific subrule keywords, (2) allowing keyword recognition without using this subrule, (3) not allowing keyword recognition. Using the first strategy (which is the one used by the system) and the second one, grammars allow the recognition of keywords. The main difference between these two strategies lies in the way that the recognition is carried out. Using the first one (see Figure 1), the initial rule of the grammar contains four subrules: order, sentence, request and keywords. The subrule for keywords includes all the words related to the control a particular appliance. For example, keywords related to the TV may be concerned with the place where this appliance is (e.g. living room or kitchen), the attribute or characteristic the user wants to change (e.g. volume or channel) and the action to be performed with the appliance (e.g. switch on or turn off).

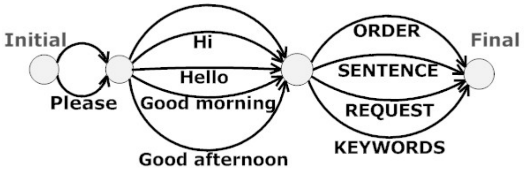


Fig. 1. Initial rule of ASR grammar including a subrule for recognition of keywords

This strategy is the most suitable for users who do not provide all the data required to perform an action with an appliance (see Section 2.1). If any data is missing to perform the action, the dialogue manager of the system prompts the user for the missing information. In this case, the user can provide just the missing data, that is, he does not need to utter the entire order. The advantage of this strategy is that the interaction is more comfortable for the user, particularly if the orders are complex and long. For example, if when processing the order:"*Set the temperature of the washing machine of the laundry room to thirty degrees*", the system does not understand the room where the appliance is located, *Mayordomo* may prompt "Where?" and the user may answer "In the laundry room".

Using the second strategy for ASR (see Figure 2), grammars also allow recognition of keywords. However, in this case, grammars do not use the specific subrule for keywords and all the elements of the subrules order, sentence and request are optional. An optional element in a rule means that it can be provided or not by the user, thus, providing it is not necessary to trigger the rule. For example, the subrule order has three elements: verb prep (subrule with a list of elements such as *switch on*), object (i.e. *the light*) and where (subrule with the rooms, for instance, *of the kitchen*). In this rule, if the element where is optional, the phrase "*Switch on the light*" will trigger the rule and will be recognized by the ASR. Therefore, this strategy permits all kinds of combination of words, resulting in a greater number of ASR errors as it allows incorrect combinations of words, for example, "*Hello washing off hall*".

Using the third strategy for ASR, the initial rule of the grammar is the same as in the second strategy. However, it does not allow optional elements in the subrules. Thus, this strategy is not advisable when the sentences are complex and long.

Speech Understanding. Speech understanding is based on what we have noted as an "*action*". In our application domain, an action consists of four fields of data: *room*, *appliance*, *attribute*, and *value* (see Table 1). Using these four elements, the system can

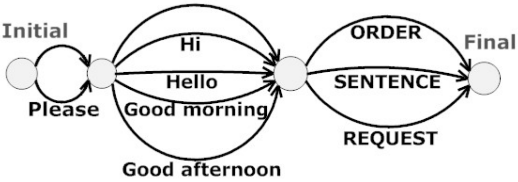


Fig. 2. Initial rule of ASR grammar without subrule for keywords

execute a particular order on an appliance, or provide the information requested by the user.

To implement the speech understanding process, we employ a method that searches in the recognized sentence for the four fields of data in the action concept. To search for the room element, the method tries to find in the sentence names of existing rooms in the house. The system is able to detect that the user refers to all the rooms in the environment.

To search for the *appliance* element, the system proceeds in a similar way, looking for the names of the existing appliances in the home. Moreover, since the user can refer to the name of the appliances in an abbreviate way (e.g., “music” instead of “piped music”), the system searches for fragments of appliances’ names in the recognised sentence.

Mayordomo proceeds in a similar way to find the item attribute, that is, it tries to locate in the recognised sentence each attribute corresponding to the appliances in the house. If an attribute is found it then considers whether an appliance has been mentioned. If it is not found, the system assumes that the user has omitted it, and it looks for appliances which have such attribute. If just one appliance is found, the action is performed on that appliance. However, if the attribute is associated with several appliances, such as the volume attribute, which can refer to television or the piped music, the system prompts the user for the target appliance.

Table 1. Description of fields in the action concept

Room	Room containing the appliance. This information is necessary to distinguish, for example, which lights have been lit
Appliance	Particular appliance on which the action is performed
Attribute	Characteristic of the appliance that is affected by the action
Value	New value for the attribute

Finally, the Semantic Analyser of the system (see Figure 3) searches for the element value following the same procedure, that is, the module tries to find in the recognised sentence names of possible values associated with attributes For example, since some attributes are numeric (in the range 0–10), the system tries to find numbers in that range within the recognized sentence.

We have observed that users often utter sentences omitting attributes and pronounce some values associated with that attributes. For example, in the sentence: “Switch on the lights”, the attribute is “State” but it does not appear explicitly in the sentence. To analyze these utterances, the system looks for verbs which represent omitted values and/or attributes. In the previous example “Switch on” is related with “State”.

To determine whether the user is asking a question or ordering an action to be carried out on an appliance, the system analyzes the recognized sentence. If it finds any word beginning with “*wh-*” or any conjugation of the verb “*to be*” in present tense, it is assumed that the user has made a question. More specifically, if the word is “*what*” or “*which*”, it is assumed that the user is asking for the value of an attribute of a particular appliance in a particular room. If the word is “*where*” it is assumed that the user is requesting information about places where appliances are located.

Dialogue Management. Once the analysis of the sentence is finished, the dialogue manager must decide what will be the answer to be generated by the system. In particular it must determine whether to provide information requested by the user or perform a specific action on an appliance (Fig. 4). To do this it checks if there is any lack of information in the recognized sentence. This lack of information can refer to at least one of the four types of data related with the action concept mentioned above. If there is no data missing and the user is requesting information, the dialogue manager calls the module *Provide Information*, which organises in well-formed sentences the information to be provided to the user.

If the user wants to carry out an action, the dialogue manager calls the module *Perform Action* which executes the action. If there is any lack of information, the dialogue manager decides which appropriate question must be generated in order to obtain that data from the user (see Figure 3). Whenever an action is performed, an entry is made in the log file containing data on date and time of the action.

The *Perform Action* module changes, if necessary, the four types of data discussed above (room, appliance, attribute and value). This change can affect just a specific room or all the rooms in the house. For example, if the user turns on the light in the kitchen, the field *room* is filled with the value “*kitchen*”, the appliance field is filled with the value “*light*”, the field attribute is filled with the value “*state*” and the attribute value is filled with “*on*”. These changes are made as well in the system’s memory.

User: Switch on the light
System: What light? The ceiling lamp or the floor lamp?
User: Ceiling lamp.
System: You have changed to on the state of the ceiling lamp in the living room.

Fig. 3. System request for missing data in the concept “action”

The dialogue manager needs to know characteristics of the house (i.e. rooms and appliances) with which it interacts. These characteristics are stored in configuration files for the house and for each appliance. The file for the house contains a generic description of the house, including number of rooms and name for each room. These

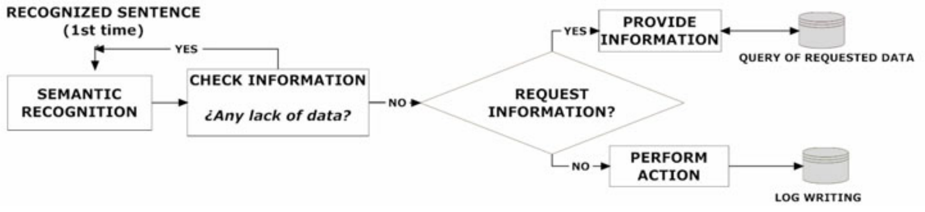


Fig. 4. Diagram of the process followed by the dialogue manager

names are recognized by the ASR and understood by the Semantic Analyser. Each room has associated a set of appliances installed in it, which are represented identifiers which are handled by the dialogue system.

The configuration file for each appliance contains information about the functionality of the appliance as well as details about attributes or characteristics and possible values for each attribute. For example, for the TV the system uses a configuration file that contains attributes such as *volume* and *channel*, as well as possible values for these attributes.

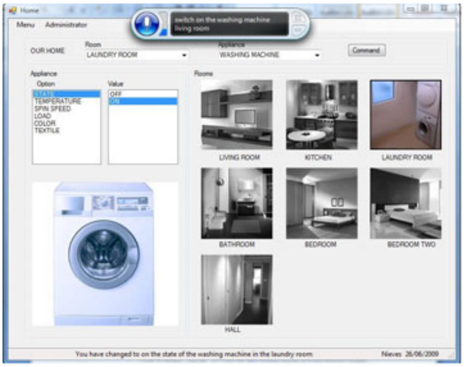
Sentence generation. As discussed in the previous section, the dialogue manager is responsible for determining the next action to be performed by the system. There are two possibilities: either oral response for the user, or execution of an action on an appliance. For example, if the user has asked the question: "*Where are the lights on*?", the system responds: "*The lights are on in the kitchen and in the living room*". However, if the user makes an order the dialogue manager executes the order and requires the generation of a confirmation message. For example, if the order is:"*Turn up the volume of the TV in the living room*", the system responds, "*You have changed to five the volume of the TV in the living room*".

To generate responses, *Mayordomo* uses a set of patterns that are instantiated with different values depending on the appliance, room, attribute and value involved. For instance, the pattern for the previous example is: "*You have changed to (value) the (attribute) of the (appliance) in the (room)*".

Speech synthesis. The system uses speech synthesis (TTS) to communicate verbally with the user, employing as input the sentences in text format created by the module for sentence generation. For example, this technique is used at the beginning of the interaction to provide a welcome message to the user and request authentication. It is also used to prompt the user to fill in missing information in order to execute a particular order and to inform the user about the status change on the required appliance.

2.2 Graphical Interaction

In order to ease interaction with the appliance to a wider range of potential users, *Mayordomo* provides a GUI interface that includes a series of buttons.



switch on the washing machine
living room

Room: LAUNDRY ROOM

Appliance: WASHING MACHINE

Option: STATE

Value: ON

You have changed to on the state of the washing machine in the laundry room

Nieves 26/06/2009

Fig. 5. GUI interface of Mayordomo

The interface (see Figure 5) has a status bar, and a text field which displays each system's response. This status bar can be very useful in case users want to interact with the system in noisy places where understanding the messages generated by TTS might be difficult. The interface also provides a command prompt that allows the user to communicate with the system in text format. This mode of interaction is designed for scenarios in which the user has to interact with noisy environments and using oral mode is not a good option. In this case, the absence of errors of ASR generates better system behaviour.

3 Conclusions and Future Work

This paper has presented the current status of the implementation of a multimodal dialogue system called *Mayordomo*, the purpose of which is to interact with appliances in a house. To optimize this control it is assumed that the system operates in an AmI environment. The multimodality allowed by the system provides two methods of interaction with appliances (oral and graphic). The system has been designed to be scalable and support a number of kinds of appliance using configuration files. The system distinguishes between different types of users, allowing certain functions to only administrators and parental control that excludes certain users of a specific set of appliances.

Future work will focus on the design and implementation of a middleware layer to represent the environment and let us handle the interaction with real appliances. At the time of writing this interaction is carried out using appliances which are represented by means of software objects. To do this we plan to implement a blackboard structure [8,2,9]. The system will interact with real appliances by means of this layer in order to change their state and get information about their current state.

References

1. Aghajan, H., Lopez-Cozar, R., Augusto, J.C.: Human-centric Interfaces for Ambient Intelligence. Academic Press, London (2010)
2. Haya, P.A., Montoro, G., Alaman, X.: A Prototype of a Context-Based Architecture for Intelligent Home Environments. CoopIS/DOA/ODBASE 1, 477–491 (2004)
3. Augusto, J.C.: Ambient Intelligence: Basic Concepts and Applications. Communications in Computer and Information Science, vol. 10, pp. 14–24. Springer, Heidelberg (2008)
4. Augusto, J.C., McCullagh, P.: Ambient Intelligence: Concepts and Applications. Int. J. Computer Science and Information Systems 4, 1–28 (2007)
5. Ramos, C., Augusto, J.C., Shapiro, D.: Ambient Intelligence – The Next Step for Artificial Intelligence. IEEE Intelligent Systems 23(2), 15–18 (2008)
6. IST Advisory Group (2001). The European Union Report, Scenarios for Ambient Intelligence (2010)
7. Garcia-Herranz, M., Haya, P., Esquivel, A., Montoro, G., Alaman, X.: Easing the Smart Home: Semi-automatic Adaptation in Perceptive Environments. Journal of Universal Computer Science 14(9), 1529–1544 (2008)
8. Alaman, X., Haya, P., Montoro, G.: El proyecto InterAct: Una arquitectura de pizarra para la implementacion de Entornos Activos. In: Proc of. Congreso de Interaccion Persona-Ordenador (Interaccion 2001), Salamanca, Spain, pp. 72–73 (2001)
9. Arroyo, R.F., Gea, M., Garrido, J.L., Haya, P.A.: Development of Ambient Intelligence Systems Based on Collaborative Task Models. Journal of Universal Computer Science 14(9), 1545–1559 (2008)

Integrating Aggregation Strategies in an In-Home Domain Dialogue System

Pablo Gervás[1], Gabriel Amores[2], Raquel Hervás[1], Guillermo Pérez[2], Susana Bautista[1], Virginia Francisco[1], and Pilar Manchón[2]

[1] Universidad Complutense de Madrid, 28040 Madrid, Spain
[2] Universidad de Sevilla, 41004 Sevilla, Spain

Abstract. This paper presents the integration of a natural language generation system onto an In-Home Domain Dialogue System to achieve fluent, non-redundant verbal descriptions of the state of the environment. Three important contributions are brought together in this integration: an in-depth study of aggregation strategies preferred by users in the In-Home Domain, a fully operational dialogue system, and a natural language generation system capable of implementing the required aggregation strategies. The integration is validated by means of acceptance tests with human evaluators. In this paper we show how the aggregation strategies remove redundancies and provide a description that is assigned higher scores by human evaluators than prior descriptions.

1 Introduction

Existing dialogue systems developed for the interaction with home automation environments achieve good performance when understanding simple commands ("turn on the light in the bathroom") and reporting to the user simple data concerning the state of the system ("the light in the sitting room is off"). This is due to the application of powerful solutions for speech recognition, language parsing, dialogue management, and text-to-speech synthesis, together with simple solutions for the generation of individual sentences. However, when having to report complex data about the state of the system (for instance, when asked to describe the state of all devices in a given room, or in the whole house), those systems face a more complex challenge involving discourse planning and aggregation. Thus, a trivial question by the user such as: *"What lights are on in the house?"* could receive several answers depending on the natural language generation capabilities of the system:

- *"There is a light on in the living room. There is a light on in the kitchen."*: enumeration of sentences generated individually.
- *"There is a light on in the living room and there is a light on in the kitchen."*: coordination of sentences generated individually.
- *"There are lights on in the living room and in the kitchen."*: natural language generation with efficient aggregation capacity.
- *"There are two lights on, one in the living room and another one in the kitchen."*: natural language generation with natural aggregation capacity.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 499–506, 2010.

As in all language generation tasks, the problem becomes one of selecting which aggregation strategy to use. The decision will depend greatly on the particular domain and the context of use. This paper focuses on the domain of system–user dialogues in a home automation environment, as exemplified in the MIMUS (**M**ult**IM**odal, **U**niversity of **S**eville) system [1]. The MIMUS system is linked to the TAP [2] natural language generator, extended for this purpose with a specific module for syntactic aggregation. The aggregation strategies implemented have been extrapolated from the results reported in a previous experiment carried out over MIMUS [3].

2 Previous Work

2.1 Aggregation

A review of the literature on aggregation [4,5,6,7] clearly points out that there is no agreement on its definition or where to place it in the generation process. Albeit thorough attempts have been made to come up with a core definition [8] and a standard architecture [9], conceptual problems arise.

For the purpose of this project, aggregation is conceived of as a process which removes redundant information from a text when it can be inferred or retrieved from linguistic sources (the remaining text), from computational sources (ontology), or pragmatically (using common knowledge).

There are different types of aggregation: conceptual, discourse, semantic, syntactic, lexical and referential. In this work, we will focus on syntactic aggregation, understanding it as the process of combining sentences by means of syntactic rules. In most cases syntactic aggregation involves aggregation of subjects or predicates. Aggregation is achieved by means of coordination and reduction. Reduction is a grammatical process whereby the structure of the sentence is abridged to avoid the redundancy of expressions, as done, for instance, by means of ellipsis.

2.2 MIMUS: A Multimodal and Multilingual Dialogue System for the Home Domain

MIMUS [1] is a multimodal and multilingual dialogue system for the In–Home scenario which allows users to control some home devices by voice and/or clicks. MIMUS follows the Information State Update (ISU) approach to dialogue management [10], and was developed under the EU–funded TALK project.[1]

The system has a symmetric architecture that allows both the input and the output to be presented in graphical, voice or mixed (voice plus graphical) modalities. Additionally, the user may interact dynamically in English and Spanish. MIMUS is made up of a series of collaborative agents that cooperate and communicate among them under the Open Agent Architecture (OAA) [11] framework. The core module in MIMUS is the Dialogue Manager (DM), an agent that is linked to a Natural Language Understanding (NLU) module and to a Natural Language Generation (NLG) module.

[1] Talk Project. Talk and Look: Linguistic Tools for Ambient Linguistic Knowledge. 6th Framework Programme. 2004. `http://www.talk-project.org`

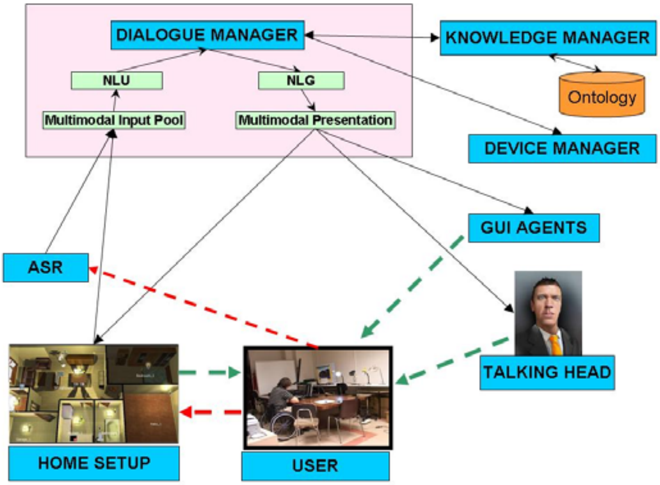


Fig. 1. MIMUS Architecture

Thus, dialogue interactions may be triggered by the semantic information provided by the user or by the dialogue expectations generated by the dialogue manager itself.

For the purpose of this paper, the challenges arise when MIMUS has to satisfy requests presented by the user for information on the state of the system. In the current configuration, these requests can be of four types:

REQ_DEVICES: refers to enquiries about which devices satisfy a set of restrictions, such as *"Which devices are switched on in the kitchen?"*

REQ_QUANTITY: requests about the quantity of devices satisfying specific restrictions: *"How many devices are on in the kitchen?"*

REQ_LOCATION: requests about the location of specific devices: *"Where are lights on?"*

REQ_STATE: requests regarding the state of specific devices: *"How is the light in the kitchen?"*

REQ_EXIST: Yes/No questions related to whether devices with certain restrictions exist in the house: *"Is there any light on in the bedroom?"*

In each of these cases, there is a significant amount of information to be conveyed as text. Each type of request also introduces particular constraints on the form of the reply.

2.3 TAP: A Text Arrranging Pipeline

TAP [2] is a framework for constructing natural language generators that convert an initial specified discourse input into text. The input must be specified in terms of a discourse, considered as an ordered sequence of messages. For each of these messages, TAP assigns a syntactic form in which the conceptual content of the message is to be realized. This syntactic form may be different for different languages. This task is known as *syntactic choice*. In messages obtained in this way mentions of referents must have been assigned specific referential expressions or *references*. TAP also assigns lexical forms to the various elements contained in a message (mainly actions and references). This task is known as *lexicalisation* or *lexical choice*.

The final task carried out by TAP is one of *surface realization*. This involves obtaining a syntactic realization of each message in the particular language specified, and putting together the realizations of individual messages into a text rendition of the linguistic content.

The TAP project originally distributed the tasks outlined above into the following modules. A Sentence Planner module is in charge of instantiating syntactic forms and lexicalization. A Surface Realizer module undertakes the subtask of converting the available data into correct sentences and putting together these realizations of individual messages into a text rendition of the linguistic content.

When instantiating the TAP framework to provide a generator for a given task, specific instantiations of these modules may be required to embody task-specific solutions. Details of the instantations of TAP to operate with the MIMUS system are given below.

3 Integrating TAP with MIMUS to Provide Aggregation in the In-Home Domain

3.1 Instantiating the TAP Framework for the In-Home Domain

The MIMUS system generates a feature–value information–state representation as specification of the reply to be given to user requests. The information contained in this representation does not include details as to the syntactic form to be employed. For the work on this paper, TAP was instantiated with the following syntactic choices:

- for REQ_DEVICES and REQ_STATE, copulative sentences are used (*"The light is on." / "La luz está encendida."*)[2]
- for REQ_QUANTITY, REQ_LOCATION and REQ_EXIST, existential sentences are used (*"There is a light on in the kitchen." / "Hay una luz encendida en la cocina."*)

References to devices and rooms are built as simple noun phrases using as common nouns the name of the corresponding concept. As it is assumed that the user is aware of the existence of both devices and rooms, definite articles are always employed.

For this particular purpose there was no need to apply lexical choice (system chooses between more than one possible way of lexicalizing each concept), so a simple process of lexicalization based on a dictionary look-up is applied.

3.2 Extension of the TAP Generator to Include Sentence Aggregation Functionality

In order to extend the functionality of TAP to include sentence aggregation, a Sentence Aggregator module was added to the general architecture. This module is placed within

[2] In the case of Spanish, the *estar* verb form must be employed in copulative sentences.

the pipeline as an intermediate stage between the Sentence Planner and the Surface Realizer. Its role is to revise the material generated so far, aggregating or omitting elements to ensure the final result is as natural and compact as possible. The relative position of the Sentence Aggregator with respect to other modules was chosen based on the fact that the decisions on whether or not to carry out syntactic aggregation require the syntactic and lexical form of messages to be decided (which is done by the sentence planner). As aggregation decisions will affect the final realization of the discourse, they must be taken before the final realization is produced by the surface realizer.

The Sentence Aggregator module receives a discourse as input and outputs another discourse where certain messages may have been fused together into a composite message, and certain elements of given messages may have been annotated as to be omitted during realization. The Sentence Aggregator module operates by progressively processing all messages in the received discourse, and grouping them together into lists of messages to be aggregated. The decisions of whether messages can be aggregated or not are taken based on the information on syntactic structure specified for the messages in the input discourse, cross-referred with information on co-occurrence of elements between adjacent messages. The co-occurring elements that are currently considered are verbs, subjects, attributes, and prepositional adjuncts describing location. Elements describing the state of devices are also considered. These take different forms in Spanish—where they are realized as past participles of the form *encendida/apagada*—and English—where they are realized as *on/off*, but are handled by the system in a similar way (though issues of number/gender agreement with the subject must be taken into account in the case of Spanish).

4 Evaluation

In order to evaluate the impact of using new aggregation strategies in the NLG module in MIMUS, a survey was conducted which compared the outputs of the original and the new NLG modules for the same set of interactions.

Evaluators first watched a video showing the overall functionality of MIMUS to familiarize themselves with the system. Then, they were handed a questionnaire with precise instructions. The questionnaire contained 26 queries to the system given a particular scenario in the house, e.g. a number of devices have been switched on in several locations in the house. All system responses should involve some kind of aggregation strategy. The original MIMUS NLG module (System A), did not include any aggregation strategy, while the new NLG module implemented on TAP (System B) did. System A's reply was always shown before System B's. The effect of consistent prior presentation of one option before the other was not considered relevant in this experiment.

A sample query is: *"Where are the lights on?"* / *"¿Dónde están encendidas las luces?"* and the replies for systems A and B are:

- System A's reply: *"There are four lights in the bathroom."* / *"Hay cuatro luces en el baño."*
- System B's reply: *"There are lights on in the bathroom and in the hall. There are dimmer lights on in the living room and in the kitchen."* / *"Hay unas luces*

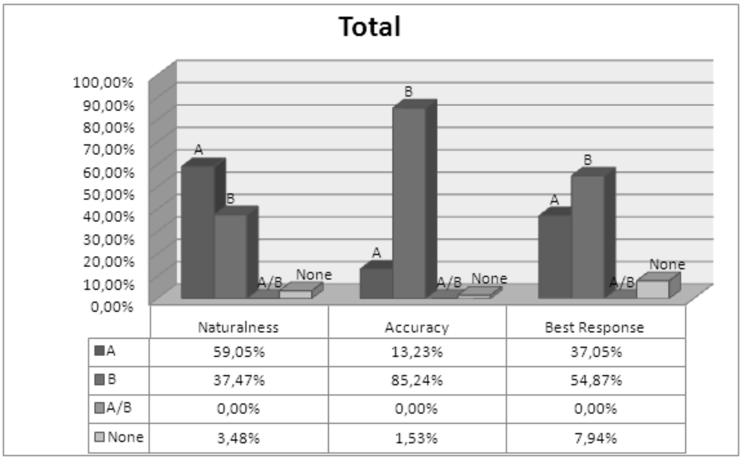
encendidas en el baño y el vestíbulo. Hay unas lámparas regulables encendidas en el salón y en la cocina.”

For each query in the survey, the informant was asked to reply to the following questions:

1. Which response sounds more natural?
2. Which response is more accurate?
3. In general, which one would you choose as best?
4. Assign a score to each response, ranging from 1 (worst) to 5 (best)
5. In your opinion, what should have been the ideal response?

A total of 28 surveys were conducted. Most informants were undergraduate students with no prior knowledge of the system or dialogue applications in general.

The results of these surveys are shown in Table 1 and in Figure 2. The evaluators considered more natural in general the responses of System A than the responses of System B. The responses of System B are considered the most accurate in most cases. As System B gets the best responses, it appears that accurate responses are valued more highly by evaluators than natural ones. The average score assigned to the responses of System A is 2.9. The average score assigned to the responses of System B is 3.74.



	Naturalness	Accuracy	Best Response
■A	59,05%	13,23%	37,05%
■B	37,47%	85,24%	54,87%
■A/B	0,00%	0,00%	0,00%
□None	3,48%	1,53%	7,94%

Fig. 2. Percentage of enquiries in which the responses of systems A or B have been considered as more natural, more accurate or better

5 Discussion

The results of the evaluation presented in Section 4 show that the answers of the system with aggregation (System B) have a higher score for Best Response in all cases. Although these results are not statistically significant, they represent a positive tendency of most subjects to score the most accurate responses in System B higher than those in System A.

Table 1. Percentages of informants that attributed Naturalness (Nat), Accuracy (Acc) and Best Response (Best) to responses from systems A and B, detailed over type of request

Request	Nat A	Nat B	Acc A	Acc B	Best A	Best B
REQ_DEVICES	17.86	**79.29**	1.43	**98.57**	1.43	**98.57**
REQ_LOCATION	44.56	**49.74**	6.22	**91.19**	22.80	**67.36**
REQ_QUANTITY	**92.03**	6.52	17.39	**82.61**	**65.22**	28.26
REQ_STATE	**90.51**	7.30	32.12	**64.96**	**65.69**	27.01
REQ_EXIST	**64.29**	33.93	12.50	**87.50**	25.00	**67.86**

In terms of accuracy, the evaluators always consider that the responses generated by System B are more accurate than those generated by System A, which leads to the conclusion that aggregated responses are always more accurate.

In those types of queries in which System B is more precise and more natural the best answer chosen by the evaluators is also the answer generated by System B as expected. However, for REQ_QUANTITY and REQ_STATE requests evaluators selected as best answers those generated by System A (showing a preference for natural over accurate responses), while for REQ_EXIST queries evaluators considered that the best responses were the ones generated by System B (showing a preference for accurate over natural responses).

An analysis of the last part of the survey, where evaluators were asked to provide an example of what they would have considered an ideal reply, provides clues to understand why some of the aggregated responses were considered less natural. Natural responses preferred by the users include features that were not taken into consideration (some of them not directly related to aggregation): more radical use of ellipsis ("In the bathroom and in the kitchen"), ommission of location information when quantity of devices is requested and their number is large, modifications in the order of presentation of elements within the sentence, or use of an initial sentence indicating the total number of devices involved to introduce subsequent descriptions.

In conclusion, we can say that aggregation generally was an improvement in the responses given by the MIMUS system although an effort should be made to improve the naturalness of the reponses.

6 Conclusions and Future Work

This paper has addressed the issue of aggregation in natural language generation in spoken dialogue systems in the context of MIMUS, a home automation environment. Reported evaluation results suggest that adding aggregation in the NLG module has increased precision in the responses generated by the system but more work must be done in order to get not only more precise but also more natural responses. To achieve this aim, as future work we will focus on improving the aspects identified as weak during the evaluation, with particular attention to the additional aspects identified as preferred by users for more natural responses.

Acknowledgements. This work has been funded by the Spanish Science and Education Ministry under projects GALANTE (TIN2006-14433-C02-01) and GILDA (TIN2006-14433-C02-02).

References

1. Pérez, G., Amores, G., Manchón, P.: A Multimodal Architecture for Home Control by Disabled Users. In: Proceedings of IEEE-ACL Workshop on Spoken Language Technology (SLT), pp. 134–137 (2006)
2. Gervás, P.: TAP: a Text Arranging Pipeline. Technical report, Natural Interaction based on Language Group, Universidad Complutense de Madrid, Spain (2007)
3. Florencio, E., Amores, G., Pérez, G., Manchón, P.: Aggregation in the in-Home Domain. Procesamiento del Lenguaje Natural 40, 17–26 (2008)
4. Dalianis, H.: Aggregation in Natural Language Generation. Computational Intelligence 15 (1999)
5. Wilkinson, J.: Aggregation in Natural Language Generation: Another look. Co-op Work Term Report. Technical report, Dept. of Computer Science, University of Waterloo (1995)
6. Shaw, J.: Clause Aggregation Using Linguistic Knowledge. In: Proceedings of the 9th International Workshop on Natural Language Generation, pp. 138–147 (1998)
7. Cheng, H.: Experimenting with the Interaction between Aggregation and Text Structuring. In: Proceedings of the ANLP NAACL 2000 Student Research Workshop, pp. 1–6 (2000)
8. Reape, M., Mellish, C.: Just What is Aggregation Anyway? In: Proceedings of the 7th European Workshop on Natural Language Processing, pp. 20–29 (1999)
9. Cahill, L., Reape, M.: Component Tasks in Applied NLG Systems. Technical report, Information Technology Research Institute Technical Report Series (1999)
10. Larsson, S., Traum, D.R.: Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit. Nat. Lang. Eng. 6, 323–340 (2000)
11. Cheyer, A., Martin, D.: The Open Agent Architecture. Autonomous Agents and Multi-Agent Systems 4, 143–148 (2001)

A Methodology for Learning Optimal Dialog Strategies*

David Griol[1], Michael F. McTear[2], Zoraida Callejas[3],
Ramón López-Cózar[3], Nieves Ábalos[3], and Gonzalo Espejo[3]

[1] Dept. of Computer Science, Carlos III University of Madrid, Spain
dgriol@inf.uc3m.es

[2] School of Computing and Mathematics, University of Ulster, Northern Ireland
mf.mctear@ulster.ac.uk

[3] Dept. of Languages and Computer Systems, CITIC-UGR, University of Granada, Spain
{zoraida,rlopezc}@ugr.es, {nayade,gonzaep}@correo.ugr.es

Abstract. In this paper, we present a technique for learning new dialog strategies by using a statistical dialog manager that is trained from a dialog corpus. A dialog simulation technique has been developed to acquire data required to train the dialog model and then explore new dialog strategies. A set of measures has also been defined to evaluate the dialog strategy that is automatically learned. We have applied this technique to explore the space of possible dialog strategies for a dialog system that collects monitored data from patients suffering from diabetes.

Keywords: dialog strategy, dialog simulation, dialog management, dialog systems.

1 Introduction

The application of statistical approaches to dialog management has attracted increasing interest during the last decade [1]. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. The final objective is to develop dialog systems that have a more robust behavior and are easier to adapt to different user profiles or tasks.

The success of these approaches depends on the quality of the data used to develop the dialog model. Considerable effort is necessary to acquire and label a corpus with the data necessary to train a good model. A technique that has currently attracted an increasing interest is based on the automatic generation of dialogs between the dialog manager (DM) and an additional module, called the user simulator, which represents user interactions with the dialog system [2].

A very important application of the simulated dialogs is to support the automatic learning of optimal dialog strategies. Large amounts of data are required for a systematic exploration of the dialog state space and corpora of simulated data are extremely valuable for this purpose, given the costs of collecting data from real users.

* This research has been funded by the Spanish Ministry of Science and Technology, under project HADA TIN2007-64718 Adaptive Hypermedia for Attention to Different User Types in Ambient Intelligence Environments.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 507–514, 2010.

In this paper, we present a technique for learning optimal dialog strategies. Our technique is based on the use of a statistical dialog manager that is learned using a dialog corpus for the specific task. A dialog simulation technique is used to automatically generate the data required to learn a new dialog model. We have applied our technique to explore dialog strategies for a dialog system designed to collect monitored data from patients suffering from diabetes. In addition, a set of specific measures has been defined to evaluate the new strategy once new simulated data is used to re-train the dialog manager. The results of the evaluation of a dialog manager developed for this system show how the variability of the dialog model is increased by detecting new dialog situations that are not present in an initial model and new valid answers for the situations that were already present.

2 The DI@L-Log System

DI@L-log is a spoken dialog system which acts as a voice logbook to collect home monitored data from patients suffering from Type-2 diabetes [3]. The data collected by the system are the patient's weight, blood pressure (systolic and diastolic values) and sugar levels. The system validates and analyses the data, providing some immediate feedback to the patients regarding their current progress as well as communicating the results to doctors at the hospital who are able to review the patient's progress graphically and deal with any alerts generated by the system concerning abnormal developments.

To apply our methodology to this task, the definition of the semantics was carried out considering the information that is required to monitor the patients and inform them about their condition. Taking into account the requirements of the task, an initial strategy was designed for the DM. Figure 1 shows an example of a dialog acquired using this strategy.

S1: Welcome to the DI@L-log System. Please say or enter your four digit pin.
U1: 1356.
S2: Are you ready to give your measurements?
U2: Yes.
S3: Let's begin with your weight. How many stones?
U3: 12.
S4: I have understood 12 stones. Is it correct?
U4: Yes.
S5: What is your blood sugar?
U5: 11.
S6: I have understood 11. Is it correct?
U6: Yes.
S7: And finally, for your blood pressure you will be asked for the top number and then the lower number. What is the top number of your blood pressure?
U7: 160.
S8: I have understood 160. Is it correct?
U8: Yes.
S9: And what is the lower number?
U9: 86.
S10: I have understood 86. Is it correct?
U10: Yes.
S11: Your readings have been successfully updated. According to them, your weight is higher than last week, be very careful. Thanks for contacting DI@L-log. Goodbye.

Fig. 1. An example of a dialog for the DI@L-log task

As can be observed, three different phases are present in every dialog. Firstly, there is an identification phase in which the system asks the user about his login and password and then waits until the user says that he is ready to provide the control data (S1 and S2 system turns). Secondly, the system analyzes which data is required for the current user, taking into account that the weight and sugar values are mandatory and the blood control is only carried out for specific patients (S3 to S10 system turns). In this phase, the system requires the user to provide this data. Every item is confirmed after the user has provided its value. The user can only provide one item at a time. In the last phase, the system consults the information that the patient has provided during the current dialog and compares it with the data that is present in a database that contains the values that he provided in previous dialogs. By means of this comparison, the system is able to inform the user about his condition and provide him with instructions that take this into account (S11 system turn).

A corpus of 100 dialogs was acquired using this strategy. In order to learn statistical models, the dialogs of the corpus were labeled in terms of dialog acts. In the case of user turns, the dialog acts correspond to the classical frame representation of the meaning of the utterance. For the DI@L-log task, we defined three task-independent concepts (*Affirmation*, *Negation*, and *Not-Understood*) and four attributes (*Weight*, *Sugar*, *Systolic-Pressure*, and *Diastolic-Pressure*).

The labeling of the system turns is similar to the labeling defined for the user turns. A total of 12 task-dependent concepts was defined, corresponding to the set of concepts used by the system to acquire each of the user variables (*Weight*, *Sugar*, *Systolic-Pressure*, and *Diastolic-Pressure*), concepts used to confirm the values provided by the user (*Confirmation-Weight*, *Confirmation-Sugar*, *Confirmation-Systolic*, and *Confirmation-Diastolic*), concepts used to inform the patient about his condition (*Inform*), and three task-independent concepts (*Not-Understood*, *Opening*, and *Closing*).

3 Our Statistical Dialog Management Technique

In most dialog systems, the DM takes its decisions based only on the information provided by the user in the previous turns and its own model. This is the case with most slot-filling dialog systems, like the DI@L-log system. The methodology that we propose for the selection of the next system answer in this kind of task is as follows [4].

We consider that, at time i, the objective of the DM is to find the best system answer A_i. This selection is a local process for each time i and takes into account the previous history of the dialog, that is to say, the sequence of states of the dialog (i.e. pairs *system-turn, user-turn*) preceding time i:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\arg\max}\, P(A_i|S_1, \cdots, S_{i-1})$$

where set $\mathcal{A}$ contains all the possible system answers.

As the number of all possible sequences of states is very large, we define a data structure in order to establish a partition in the space of sequences of states (i.e., in the history of the dialog preceding time i). This data structure, that we call Dialog Register

(*DR*), contains the information attributes provided by the user throughout the previous history of the dialog. Using this data structure, the selection of the best A_i is given by:

$$\hat{A}_i = \operatorname*{arg\,max}_{A_i \in \mathcal{A}} P(A_i | DR_{i-1}, S_{i-1})$$

The selection of the system answer is carried out through a classification process, for which a multilayer perceptron (MLP) is used. The input layer receives the codification of the pair (DR_{i-1}, S_{i-1}). The output generated by the MLP can be seen as the probability of selecting each of the different system answers defined for a specific task. The *DR* defined for the DI@L-log task is the sequence of four fields related to the information that the system requires from the user (*Weight*, *Sugar*, *Systolic-Pressure*, and *Diastolic-Pressure*).

4 Our Dialog Simulation Technique

Our approach for acquiring a dialog corpus is based on the interaction of a user simulator and a DM simulator [5]. Both modules use a random selection of one of the possible answers defined for the semantics of the task (user and system dialog acts). At the beginning of the simulation, the set of system answers is defined as equiprobable. When a successful dialog is simulated, the probabilities of the answers selected by the dialog manager during that dialog are incremented before beginning a new simulation.

An error simulator module has been designed to perform error generation. The error simulator modifies the frames generated by the user simulator once it selects the information to be provided. In addition, the error simulator adds a confidence score to each concept and attribute in the frames. Experimentally, we have detected 2.3 errors per dialog in the initial corpus of 100 dialogs acquired for the task. This value can be modified to adapt the error simulator module to the operation of any ASR and NLU modules.

The DM simulator considers that the dialog is unsuccessful when one of the following conditions take place: i) The dialog exceeds the maximum number of system turns; ii) the answer selected by the DM corresponds with a query not required by the user simulator; iii) the database query module provides an error warning because the user simulator has not provided the mandatory information needed to carry out the query; iv) the answer generator provides an error warning when the selected answer involves the use of a data item not provided by the user simulator. A user request for closing the dialog is selected once the system has provided the information defined in its objective(s). The dialogs that fulfill this condition before the maximum number of turns are considered successful.

4.1 Measures Defined for the Evaluation

We propose three measures to evaluate the evolution of the dialog strategy once the simulated dialogs are used to reestimate it. These measures are calculated by comparing the answer automatically generated by the DM for each input in the test partition with regard to the reference answer annotated in the evaluation corpus. This way, the evaluation is carried out turn by turn. These three measures are: i) *%strategy*: the percentage of answers provided by the DM that exactly follow the initial strategy defined for the task; ii) *%coherent*: the percentage of answers provided by the DM that are coherent with the current state of the dialog although they do not follow the original strategy; iii) *%error*: the percentage of answers provided by the DM that would cause the failure of the dialog.

The measure *%strategy* is automatically calculated, evaluating whether the answer generated by the DM follows the set of rules defined for the initial strategy. On the other hand, the measures *%coherent* and *%error* are manually evaluated by an expert in the task. The expert evaluates whether the answer provided by the DM allows the correct continuation of the dialog for the current situation or whether the answer causes the failure of the dialog. (e.g., the DM suddenly ends the interaction with the user, a query to the database is generated without the required information, etc).

5 Evaluation Results

Firstly, we evaluated the behavior of the original DM that was learned using the initial corpus of 100 dialogs acquired using the strict strategy described in section 2. A 5-fold cross-validation process was used to carry out the evaluation of this manager. The corpus was randomly split into five subsets of 253 samples (20% of the corpus). Our experiment consisted of five trials. Each trial used a different subset taken from the five subsets as the test set, and the remaining 80% of the corpus was used as the training set. A validation subset (20%) was extracted from each training set. Table 1 shows the results of the evaluation.

Table 1. Results of the evaluation of the initial DM learned for the DI@L-log task

	%strategy	*%coherent*	*%error*
System answer	96.11%	97.45%	2.55%

The results of the *%strategy* and *%coherent* measures show that the satisfactory operation of the developed DM due to the structure of the dialogs is almost the same in the complete set of dialogs of the initial corpus. The codification developed to represent the state of the dialog and the good operation of the MLP classifier make it possible for the answer generated by the manager to agree with one of the valid answers of the defined strategy (*%strategy*) by a percentage of 96.11%. Finally, the number of answers generated by the MLP that can cause the failure of the system is only a 2.55% percentage. An answer that is coherent with the current state of the dialog is generated in 97.45% of cases. These last two results also demonstrate the correct operation of the classification methodology.

Then, by employing the methodology proposed in this paper for dialog simulation, we acquired 100,000 dialogs which have been useful for the manager to deviate from the initial strategy. Table 2 summarizes the statistics of the acquisition of the simulated corpus for the DI@L-log task. A set of 11 different scenarios was defined to specify the objectives of the simulation, taking into account if the pressure values are necessary and different possibilities for the generation of errors and confidence measures. Given that the first and third phases of the dialog are always mandatory and have always the same structure, only the second phase in which the system collects the different values to monitor patients was taken into account for the simulation.

Table 2. Statistics of the corpus acquisition for the DI@L-log system

Simulated dialogs	100, 000
Successful dialogs	27,521
Different dialogs	1,573
Number of user turns per dialog	7.02

Figure 2 shows an example of a dialog from the acquired corpus. The objective defined for the dialog was to collect the weight, sugar and pressure values. The values defined in the scenario are 12, 11, 160 and 80 respectively. Confidence scores generated by the error simulator are shown between brackets. A sentence in natural language, that is equivalent to the labeled system turn selected by the DM, has been incorporated to clarify the explanation.

———————

LOGIN PHASE

———————

S1: (*Weight*)
Tell me your weight in stones?
U1: *Weight*: 12 [0.1]
S2: (*Confirmation-Weight*)
I have understood 12 stones. Is it correct?
U2: (*Acceptance*) [0.9]
S3: (*Sugar*)
What is your blood sugar?
U3: *Sugar*: 11 [0.9]
S4: (*Systolic*)
Tell me your blood systolic pressure.
U4: (*Systolic*): 150 [0.1]
S5: (*Confirmation-Systolic*)
I have understood 150. Is it correct?
U5: (*Rejection*) [0.9]
S6: (*Systolic*)
Tell me your systolic pressure.
U6: *Systolic*: 160 [0.9]
S7: (*Diastolic*)
And what is the lower number?
U7: *Diastolic*: 80 [0.9]

———————

DATA ANALYSIS - FINAL ANSWER

———————

Fig. 2. A dialog extracted from the simulated corpus of the DI@L-log task

In this dialog, the system begins asking the user about his weight. As a low confidence measure is introduced for the value provided by the user simulator in U1, the system decides to confirm this value in S2. Then, the system asks for the sugar value. The user simulator provides this value in U3 and a high confidence measure is assigned. Therefore, this value does not need to be confirmed by the system.

The system asks for the diastolic pressure in S4. An error is introduced in the value provided by the error simulator for this parameter (it changes 160 to 150) and a low confidence measure is assigned to this value. Then, the system asks the user to confirm this value. The user simulation rejects this value in U5 and the system decides to ask for it again. Finally, the system asks for the systolic pressure. This value is correctly introduced by the user simulator and the user simulator also assigns a high confidence level. Then, the system has the data required from the patient and the third phase of the dialog carries out the analysis of the condition of the patient and informs him.

Finally, we evaluated the evolution of the DM when the successful simulated dialogs were incorporated into the training corpus. A new DM model was learned each time a new set of simulated dialogs was generated. For this evaluation, we used a test partition that was extracted from the simulated corpus (20% of the samples). Table 3 shows the results of the evaluation of the DM model after the successful dialogs were incorporated into the training corpus.

Table 3. Results of the evaluation of the DI@L-log DM obtained after the dialog simulation

	%strategy	*%coherent*	*%error*
System answer	13.64%	98.84%	1.16%

Figure 3 shows the evolution of *%strategy*. It can be observed how the original strategy was modified since the measure decreases to 13.64%, thereby allowing the DM to tackle new situations and generate new coherent answers for the situations already present in the initial corpus. Due to the new learning process, the DM can now ask for the required information using different orders, confirm these information items taking into account the confidence scores, reduce the number of system turns for the different kinds of dialogs, automatically detect different valid paths to achieve each of the required objectives, etc. The values obtained for the coherent and error measures also indicate the correct performance of the enhanced DM.

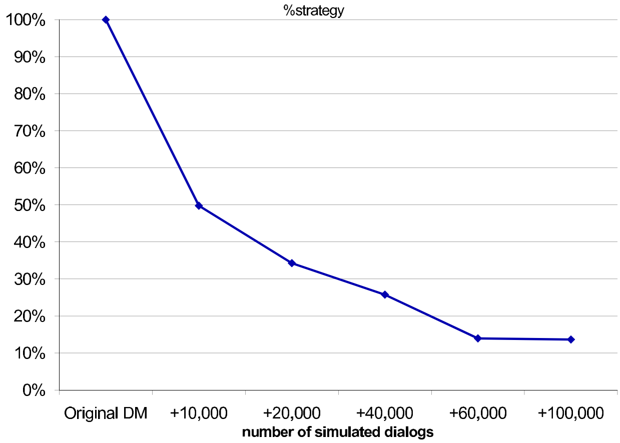


Fig. 3. Evolution of the *%strategy* measure measure with regard to the incorporation of new simulated dialogs in the DI@L-log task

6 Conclusions

In this paper, we have described a technique for exploring dialog strategies in dialog systems. Our technique is based on two main elements: a statistical dialog methodology for dialog management and an automatic dialog simulation technique to generate the data that is required to re-train the dialog model. The results of applying our technique to the DI@L-log system, which follows a very strict interaction flow, show that the proposed methodology can be used not only to develop new dialog managers but also to explore new enhanced strategies. Carrying out these tasks with a non-statistical approach would require a very high cost that sometimes is not affordable. As a future work, we are adapting the proposed dialog management for its application in more difficult domains, in which a previous plan recognition phase would be necessary.

References

1. Young, S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Technical report, CUED/F-INFENG/TR.433, Cambridge University Engineering Department, Cambridge (UK) (2002)
2. Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S.: A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. Knowledge Engineering Review 21(2), 97–126 (2006)
3. Black, L.A., McTear, M.F., Black, N.D., Harper, R., Lemon, M.: Appraisal of a Conversational Artefact and its Utility in Remote Patient Monitoring. In: Proc. of the 18th IEEE Symposium CBMS 2005, Dublin, Ireland, pp. 506–508 (2005)
4. Griol, D., Hurtado, L., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. Speech Communication 50, 666–682 (2008)
5. Griol, D., Hurtado, L.F., Sanchis, E., Segarra, E.: Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique. In: Proc. of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium, pp. 39–42 (2007)

The Structure of a Discontinuous Dialogue Formed by Internet Comments

Tiit Hennoste, Olga Gerassimenko, Riina Kasterpalu, Mare Koit,
Kirsi Laanesoo, Anni Oja, Andriela Rääbis, and Krista Strandson

University of Tartu, J. Liivi 2, 50409 Tartu, Estonia
{tiit.hennoste,olga.gerassimenko,riina.kasterpalu,mare.koit}@ut.ee,
{laanesk7,anni.oja,andriela.raabis,krista.strandson}@ut.ee
http://www.cl.ut.ee

Abstract. We are studying how a dialogue structure is established by an Internet opinion article and its anonymous comments. We are using the methodology of conversation analysis with the focus on membership categorization analysis. The study shows that the core structure of the dialogue is formed by many parallel micro-dialogues. Besides the linear micro-dialogue structure there is a structure layer which is formed by the complex category sets built by participants using membership categorization of the agents of the article as well as the commentators themselves. We investigate the strategies and the linguistic means used by the participants for creating interrelations in the complex multilayered structure.

Keywords: Internet comments, dialogue, communicative strategies, conversational analysis, membership categorization analysis.

1 Introduction

We are studying a dialogue formed by an Internet opinion article and its anonymous comments. Such texts are produced and published with the aim to achieve as many comments as possible. A big number of comments demonstrates the popularity of a portal and it will be taken into consideration by advertising agencies in concluding financial contracts. Strategies of constructing texts take into account so-called commenting values – properties of events and/or agents which probably provoke quick and emotional comments [1].

As a rule, the author of the source text and the agents of the text do not participate in the emerging dialogue. However, every commentator can write more than one comment. The communicative goal of a participant is to express his/her opinion. If s/he terminates the interaction then a conclusion can be made that s/he reached his/her communicative goal or abandoned it. If a participant gives more than one comment (that can be recognized by IP addresses) then s/he is interacting during a longer time.

We can make an hypothesis that the commentators have read through the source text as well as the previous comments. They have a possibility to react to the source text or to some comment. In this way, a dialogue is formed by occurring parallel micro-dialogues: source text – comment 1, source text – comment 2, etc. On the other hand,

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 515–522, 2010.

longer sub-dialogues can appear if a commentator gives his/her opinion connected with a previous comment. As well as we know, the structure of such discontinuous dialogues has not been analysed before.

The paper is organised as follows. Section 2 gives an overview of the analysis method – membership categorization analysis (MCA) – and describes the used empirical material. Section 3 is dedicated to the analysis. Section 4 discusses some problems and Section 5 makes conclusions.

2 The Method and Empirical Material

Our method derives from Conversation Analysis (CA) which is a method to analyse the microstructure of conversation and the coherence of dialogue established by cooperation of participants [3]. According to CA, the main mechanisms of conversation organization are turn taking, sequence organization, repair and action formation. A basic idea of CA is that conversation is organized sequentially. An important unit for sequence construction is an adjacency pair (AP) which is composed of adjacently placed and relatively ordered two turns by different speakers. The first pair part makes some second pair part relevant (e.g. question needs answer). A source text of an Internet dialogue can be considered as a statement. CA considers this action as the first part of an AP in a spoken interaction. It is expecting the second pair part, a response of the hearer.

Commentators are anonymous. Their properties (social status, age, gender, etc) are generally unknown for the investigator. Because of that, we decided to use the method of the membership categorization analysis (MCA). MCA comes from the same roots as CA and investigates how people categorize themselves and other persons in conversation [2,4,5,6]. The same person can be a member of many categories. The categories belonging together form collections (man-woman, liberal-socialist-conservative, etc). Categorization is indexical and related to the setting of conversation. People do not use ready-made categories but they change and develop them in the course of conversation. Categorization can be topic-relevant (related to the conversation topic), institution-relevant (related to the state, ideology, etc) and conversation-relevant (related to the roles of participants in interaction, e.g. commentator, moderator, etc).

Categories are inference rich. Every member of a category must have the properties of that category (e.g. there are typical properties of a young woman). Individuals end up in different ways, being members of the same categories under the application of alternative category collections.

Using categorization, a participant stresses certain properties and makes them relevant for other participants. A category makes it possible, prepares, and avoids something in the following actions of participants. At the same time, categorization carries a local role in a dialogue. When choosing a categorization, a participant relates his/her text to the previous text.

Our empirical material is an opinion text[1] (Simson: Unemployed Lose Hope or Leave Estonia) together whith its comments published on the Estonian Internet portal

[1] http://www.delfi.ee/news/paevauudised/eesti/simson-tootud-kaotavad-lootuse-ja-lahkuvad-eestist.d?id=28495983

Delfi on January, 15, 2010. The author is a journalist. The topic of the text is unemployment. Opinions of two young politicians are presented, both considering how to reduce the unemployment. The agents of the source text are Kadri Simson (a woman, a member of a left-side opposition party, the chairperson of the parliament fraction) and Taavi Rõivas (a man, a member of the liberal government party, the chairman of the financial commission of the parliament).

The unemployment is important for many people of Estonia, both parties and their leaders awake strong emotions, both of them have strong supporters as well as strong antagonists.

The source text was published at 6:55. The first comment arrived at 7:13. The total number of comments is 171. The main part of comments typically arrives during a short time after the publication of a news story. In our case, 161 comments were written during the first day and the remaining 10 during the following 5 days.

3 Analysis

3.1 Relations between the Source Text and the Comments

The source text presents opinions of two politicians. Therefore, it can be considered as a dialogue act Statement – the first pair part of an AP. In newspaper opinion articles typically two opposite opinions are presented. A commentator can choose a side and, in this way, express his/her agreement with either side or non-agreement with both sides. The number of different commentators is 127. Most of them (86%) takes a turn only once and relate themselves directly to the source text. The communicative goal of commentators is to express their opinions about the agents of the source text (Simson and Rõivas) and/or positions represented in the source text. 20% of comments are directly related to the agents (*Kadrikene, see on ..../ little Kadri, it is ....; töötult Kadrile/ from an unemployed to Kadri*). About 40% of comments are related to the source text via topic. The commentators agree or disagree with the presented positions and express their own opinion. Such comments can be considered as the second pair parts of an AP and as well as potential first pair parts of new APs.

When considering the situation from the standpoint of a dialogue, a commentator chooses the next speaker (an agent of the source text) but no conversation follows (because the agents do not participate in conversation). It results in many parallel micro-dialogues consisting from one AP (source text – comment 1, source text – comment 2, and so on).

The second group of commentators (38% of comments) relate themselves to other comments. In this way, sequences of comments appear where the first comment is directly related to the source text, the next comment is related to the previous comment, and so on. In the following example, the first comment presents a negative opinion about the goals of both parties, then the next comment argues for the opposition party and, at the same time, evaluates the previous commentator (*lollike/little fool*) who does not react. Still, it is not like a spoken face-to-face dialogue because the addressee of the comment 2 does not react to the turn directed to him/her.

kuule lollike Naljakas, 15.01.2010 07:20

mida sa siin hommiku vara plõksid. Otse loomulikult on see ideaalne idee keskerakonna poolt. Vastik juba vaadata seda ilkumist, et kui keskerakond siis ükski mõte ja idee ei sobi. Kuhu on meid reformierakond viinud või mida andnud. Ise olles ettevõtja võin öelda kohe, et kui selline seadus sotsmaksude kohapealt vastu võetaks, annaks ma tööd 6 le inimesele.Aga kuna sots maksud on nii kõrged, viiks see mind hetke olukorras pankrotti.

listen you, little fool Funny 15.01.2010 07:20

what do you twaddle here early morning. Naturally, the idea of the opposition party is perfect. It is unpleasant to see the opinion that if the opposition party then no thought or idea is acceptable. Where has the government party brought us or what it has given to us? Being an undertaker, I can say that if the law about social taxes would be taken, then I would give a job to 6 persons. At the moment I can't do so because the taxes are so high that I would go into bankruptcy.

The third group of comments simultaneously reacts to the source text as well as some of the previous comments. 18 commentators take a turn more than once, relating their comments to different turns at different moments. The most active commentator takes a turn 14 times.

When comparing the structure of the formed discontinuous dialogue with the spoken face-to-face interaction, we can state the following. If a participant reacts to an opinion presented in the source text and gives his/her own opinion then relations between the two turns are formed, following social norms of building APs in spoken interactions, even though the participants do not have responsibility for the maintenance of the conversation. The source text is like the first part of an AP, and many comments its second parts and/or the first parts of following APs because they expect following comments as the second parts. As a result, we witness a "broom" structure of many parallel micro-dialogues.

3.2 Categorization as a Communicative Strategy of Participants

The main strategy used by participants of such a dialogue where the participants are interchanging their opinions is categorization of the world.

The source text introduces some categorizations: unemployed/old people/undertakers, leavers to abroad/unemployed who have lost hope and stay in the home country without work, opposition/government (party).

After that, commentators start to categorize themselves and other people. The main **topic** of our source text is unemployed people leaving Estonia or losing hope. It implies various sets of people on the basis of topic-relevant categorization.

The first general set of categories is Homeland. The following categorizations are used: local people/people living abroad, people staying here/leaving for abroad, people who valuate home/nomads, people who valuate good life abroad/patriots.

The second set of categories is Unemployment. The following categorizations are used which can be called as a Way of acting: offenders/people who discuss, scrappers / people holding together, (active) people forging ahead/(passive) whiners, participants/observers, informed people/other, neutral/affective people, winners/losers, old/young, educated/uneducated, wise/fool.

Institutional-relevant categorization is related to the topic. Two sets of categories are used: Party and ideology, and State and folk.

The following categories belong to the first set Party and ideology: opposition /government party, politicians/non-politicians, left-side/right-side ideology. With the category set State and folk, the following categorizations are used: officials/non-officials, gradual/equal taxes, people who trust themselves/who hope that the state will help them, the state as a robber/a person who pays taxes as a victim, rich/poor people, slaves/gentry, employer/employee, unemployed/employee, waster/economical. In addition, nationality is used with the categorization of Estonians/Russians/other nationalities.

The source text does not use categorization on the basis of family. Nevertheless, this categorization is introduced in comments: people having a family/single, having/not having children, man/woman, patriarchal/non-patriarchal.

Communication-relevant categorization is explicitly expressed by one commentator who presents many comments and tries to moderate communication proposing himself/herself as an all-knowing person staying on neutral grounds versus quarellers missing some relevant points and nuances.

In the beginning of commenting process, topic-relevant categorization is mostly used (unemployment), after that the institutional-relevant categorization dominates. In the end, sub-dialogues between participants arise. Categorization is used by participants also in order to indicate relations with the authors of previous turns. Some commentators use special user names (*to X*), some of them use direct categorization of the addressees in the beginning of the comment (*hear, fool*, ...) some of them use self-categorization from the same category set as the addressee (*countryman* > *townsman*).

We can claim that the central communicative strategy used by commentators is membership categorization. The source text presents an initial categorization of the world. The goal of commentators is to present their own picture of the world, their own confrontations. To achieve this goal, they use categorization of politicians and themselves. In this way, besides the linear micro-dialogue structure there is at least one additional structure layer in the discontinuous dialogue. The layer is formed by the complex category sets built by participants using membership categorization of the agents of the source text as well as the commentators themselves. Therefore, the coherence of turns is also structured non-linearly.

3.3 Categorization of Agents

The main agent of the source text is Kadri Simson who is set in contrast with Taavi Rõivas.

The commentators have clear positive or negative attitudes in relation to both agents. There are 25 comments to Simson and 9 to Rõivas. The difference in numbers can be explained with the fact that Simson is far more known politician. She has been in public politics since 2003 while Rõivas since 2007. Negative attitude is expressed in most comments (19 for Simson, 9 for Rõivas).

If we give marks -1 and $+1$ to negative and positive comments respectively then we can represent the collective opinion of commentators about the agents of the source text as a diagram in the Fig. 1. To Simson, all the comments were given during the first four

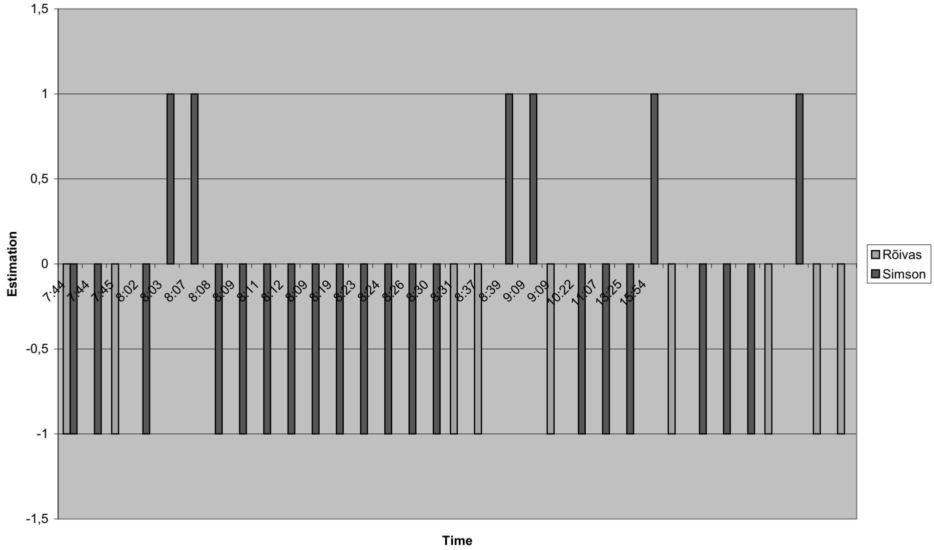


Fig. 1. "Portraits" of the Agents

hours after publication of the source text. To Rõivas, the last two comments came 7 and 9 hours later.

Simson's negative categorizations are e.g. *senseless old woman* (Simson is young!), *childish mouth, comrade Simson, unpractical little official, nice lady Kadri.* Simson is clearly categorized on the basis of her gender, age, marital status, not having children (and not as much as a political figure). On the contrary, only one comment points to Rõivas' age (*young man*); his gender and marital status are not explicitly pointed out. We can see how a woman – top-politician is determined through home and family.

At the same time there is clear difference in the means used to categorize Simson and Rõivas. Only the family name is used in negative categorizations of Taavi Rõivas but Kadri Simson is categorized in two different ways. In negative categorizations almost always her first name (*Kadri*) or diminutives (*Kadrikene*) are used. In positive categorizations, on the contrary, only the family name appears.

4 Discussion

Categorization is a complex activity, different categories can be used by a participant when constructing a turn. A certain "sharing of work" appears between the categories. Some of the categories are expressed by naming, some by activities. For example, self-naming indicates the ideological confrontation (patriot of Estonia, undertaker, a honest citizen, nationalist).

Categorization forms a complete system. All participants are using similar linguistic means (e.g. names are used in similar way).

People are using previous categorizations and take them into account in current situation. We can suppose that a social norm exists for categorization and people intuitively use it in order to act in the changing world, to determine their position, to express their attitudes. Interaction would be impossible if every participant would have his/her own, different system of categories.

However, a participant can take a temporary role in interaction. When investigating written comments (texts) and not knowing the real people, we are not able to decide whether a commentator really is a patriot in the meaning this category carries for him/her or s/he only is playing this role.

Sets of categories exist as seen before (Homeland, Unemployment, Party and ideology, etc). Each set in its turn includes sets of scales (e.g. Homeland includes scales as local people/living abroad, people staying here/leaving for abroad, etc). Using the scales, we can build "portraits" of persons in relation to every scale. Then using the "portraits" we can compose "summaries" which represent all the participants as well as the discontinuous dialogue as a whole. This informal description must be formalized in further work.

As we have seen, categorization creates relations between the source text and its comments. On the other hand, categorization can be considered as a communicative strategy of participants. Using categorization, a commentator gives positive, negative or neutral evaluations to another participant (incl. the agents of the source text) or himself/herself. One of the communicative goals of participants is creating a (collective) portrait of agents.

5 Conclusion

We are studying how an inherent dialogue structure is established by the Internet article and its comments. This structure is emerging through linguistic means used dynamically by participants. The dynamic use of linguistic means implies that interactants do not have ready-made algorithms for producing the discontinuous dialogue united by common categorization systems but they build and express the coherence and the common categories during the course of an interaction, explicating the aspects of reality which they consider relevant at this certain point.

We are using the methodology of conversation analysis with the focus on membership categorization analysis. Originally, this method was intended for the analysis of spoken coherent dialogues. We find it appropriate to use also for the analysis of a written discontinuous dialogue on Internet.

Firstly, the study shows that the core structure of the dialogue is formed by micro-dialogues consisting of two turns: the source article and its comment. Another group of comments are not necessarily associated with the source text but are directly related to some previous comment. Thus, coherent parallel sub-dialogues are formed like in the spoken conversation. In the third group, there are comments associated with more than one of previous comments. These processes result in a network of many parallel micro-dialogues.

Secondly, our analysis demonstrates that relations between turns are formed following the social norms of building adjacency pairs in spoken face-to-face interactions, even though the participants do not have responsibility for the maintenance of the conversation.

Thirdly, besides the linear micro-dialogue structure there is at least one additional structure layer. The layer is formed by the complex category sets built by participants using membership categorization of the agents of the news text as well as the commentators themselves. In this way, the coherence of turns is also structured non-linearly.

Acknowledgments

This work is supported by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science (EXCS), the Estonian Science Foundation (grant 7503), and the Estonian Ministry of Education and Research (projects SF0180078s08, EKKTT09-57, and EKKTT09-61).

References

1. Hennoste, T.: Estonian Media Climate from Spring to Autumn (2009), http://www.eurozine.com/articles/2009-11-13-hennoste-en.html
2. Hester, S., Eglin, P. (eds.): Culture in Action: Studies in Membership Categorization Analysis. Studies in Ethnomethodology and Conversation Analysis. University Press of America, Washington (1997)
3. Hutchby, I., Wooffitt, R.: Conversation Analysis. Polity Press, Cambridge (1998)
4. Lepper, G.: Categories in Text and Talk: A Practical Introduction to Categorization Analysis. Sage Publications, London (2000)
5. Sacks, H.: Lectures on Conversation. Blackwell, Oxford (1992)
6. Schegloff, E.A.: A Tutorial on Membership Categorization. Journal of Pragmatics 39, 462–482 (2007)

Using Knowledge about Misunderstandings to Increase the Robustness of Spoken Dialogue Systems

Ramón López-Cózar*, Zoraida Callejas, Nieves Ábalos,
Gonzalo Espejo, and David Griol

Dept. of LSI, CITIC-UGR, University of Granada, Spain
Dept. of Computer Science, Carlos III University of Madrid, Spain
{rlopezc,zoraida}@ugr.es, {nayade,gonzaep}@correo.ugr.es,
dgriol@inf.uc3m.es

Abstract. This paper proposes a new technique to enhance the performance of spoken dialogue systems employing a method that automatically corrects semantic frames which are incorrectly generated by the semantic analyser of these systems. Experiments have been carried out using two spoken dialogue systems previously developed in our lab: Saplen and Viajero, which employ prompt-dependent and prompt-independent language models for speech recognition. The results obtained from 10,000 simulated dialogues show that the technique improves the performance of the two systems for both kinds of language modelling, especially for the prompt-independent language model. Using this type of model the Saplen system increased sentence understanding by 19.54%, task completion by 26.25%, word accuracy by 7.53%, and implicit recovery of speech recognition errors by 20.30%, whereas for the Viajero system these figures increased by 14.93%, 18.06%, 6.98% and 15.63%, respectively.

Keywords: Spoken dialogue systems, speech recognition, spoken language understanding, user simulation.

1 Introduction

Many techniques have been published addressing the development of robust spoken dialogue systems (SDSs). Some of these work at the dialogue management level. For example, some authors propose to allow flexibility in user interaction in an attempt to handle unexpected contributions and interpret them correctly within the dialogue context [3,4]. There are also techniques that work at the speech recognition level. For instance, a number of papers propose to employ confidence scores to measure the reliability of each word in the recognised sentence [1,2]. A score lower than a certain threshold suggests a high probability that the recognised word is incorrect, in which case the system's dialogue manager usually either rejects the word or generates a confirmation prompt, thus providing some sort of mechanism for recovering from recognition errors. However, a problem with these scores is that there may be false positives and false negatives, of which the user is aware once the system rejects the word or prompts for a confirmation.

* This research has been funded by the Spanish project HADA TIN2007-64718.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 523–530, 2010.

2 The Proposed Technique

In this paper we propose a technique to increase the robustness of SDSs which is independent of the speech recogniser employed by the dialogue system, and does not use confidence scores. The technique is implemented in the robust speech understanding module shown in Fig. 1, which is comprised of the SLU component of a dialogue system and a new module that carries out frame corrections. This module receives each frame created by the SLU component and uses what we call a *correction model* (CM) to decide how to correct misunderstandings caused by speech recognition errors. The correction is carried out by replacing each incorrect frame with another that is assumed to be correct in a particular dialogue state. The output of the frame correction module is a frame, possibly corrected, that is the input for the dialogue manager of the system. Hence, some speech recognition errors may be unnoticed by the user given that the frames that finally make up the input for the dialogue manager are correct.

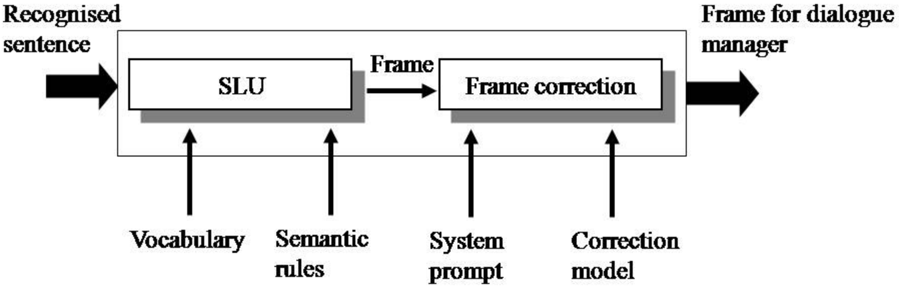


Fig. 1. Robust speech understanding module

To set up the technique for a given dialogue system, the system designers must create the appropriate correction model, which can be done by carrying out the following two tasks: creation of an initial correction model and optimisation of this model.

2.1 Creation of an Initial Correction Model

The initial correction model (ICM) stores incorrect frames generated by the SLU component of the system as it processes the input utterances. The model is comprised of tuples of the form: (T', f_R, f_O), where T denotes a prompt type generated by the dialogue system, f_R represents the reference frame associated with the sentence uttered to answer the prompt, and f_O denotes the frame obtained by the SLU component of the system as it analyses the recognised sentence. We consider that f_O is correct if it matches f_R exactly, and is incorrect otherwise. To create the correction model in the experiments we used a simple procedure that takes as its input the current prompt type (T), the reference frame (f_R) associated with the sentence uttered by the user, and the frame obtained (f_O) from the analysis of the recognised sentence. If f_R does not match f_O the procedure includes the tuple (T', f_R, f_O) into the correction model.

2.2 Optimisation of the Initial Correction Model

The second task to implement the proposed technique is to optimise the ICM to obtain the model that the frame correction module will finally use. This task can be performed by carrying out three sub-tasks: compaction, removal of inadequate tuples and expansion.

Compaction. The goal of this subtask is to make the ICM as small as possible to avoid any unnecessary processing delay. The reason is that for each input sentence uttered by the user to answer a system prompt of type T, with associated reference frame f_R and obtained frame f_O, the frame correction module will check if the tuple (T', f_R, f_O) is in the model to decide whether to correct f_O.

Removal of inadequate tuples. The second sub-task is to analyse the tuples in the already compacted ICM to prevent the frame correction module from replacing frames incorrectly. To do this we employ two models called Σ and Π, which must be created in advance by the system designers applying their knowledge about the application domain and the performance of the dialogue system. The Σ model provides knowledge about the frames used in the application domain. The Π model provides knowledge about pairs of the form: (*promptType*, *typeOfObtainedFrame*), which represent expected prompt-answer pairs in the application domain.

To remove the inadequate tuples from the ICM, the algorithm firstly analyses each tuple (T', f_R, f_O) taking into account T, f_O and the Σ model. If f_O is found to be spurious, the corresponding tuple is removed from the model to avoid causing incorrect frame replacements. Secondly, we determine whether the type of f_O matches the prompt type T taking into account the Π model. If both match the tuple is removed from the model to avoid making incorrect frame replacements. The final part of the algorithm checks that there are no tuples in the ICM with the same prompt type T and obtained frame f_O which differ in the reference frame f_R, as this would mean that the obtained frame could be corrected in several ways. Hence, if these tuples are found they are also removed from the correction model to prevent incorrect frame replacements from being made.

Expansion. The third sub-task is to expand the ICM, now compacted and free from inadequate tuples, to generalise the behaviour of the frame correction module so that it can deal appropriately with misunderstandings not observed in the training. To do this the algorithm uses a model that we call Ψ, which must be created in advance by the system designers taking into account all the possible prompt types that the system can generate. The prompt types must be grouped into classes considering as a classification criterion that all prompt types in a class must have the same expected kind of response from the user. Firstly, the algorithm for carrying out the expansion copies the tuples in the ICM to a new model CM' that is initially empty. Next, it takes into account each tuple (T', f_R, f_O) in the ICM and uses a function called *Class* to determine the class (K) of prompt types that contains T. Then it looks for tuples of the form (T', f_R, f_O) with $T' \neq T$ in CM', where T' represents each prompt type in K. If a tuple (T', f_R, f_O) is not in CM' then it is added to this model. Finally, a new correction model CM is created containing the tuples in CM'. This model will be the input for the frame correction module.

3 Experiments

The goal of the experiments was to test whether the proposed technique is useful to enhance sentence understanding (SU), task completion (TC) and the implicit recovery of speech recognition errors (IR) of two dialogue systems previously developed in our lab: Saplen [5] and Viajero [6], which employ the two-level speech recogniser that we developed in a previous study [8]. Additionally, we wished to check the effect of the technique (if any) on word accuracy (WA).

To carry out the experiments we used a user simulation technique that we developed in a previous study [7]. In order to avoid excessively long dialogues between the system and the user simulator, which would not be accepted by real users, we made the simulator cancel the interaction with the system if the total number of interactions (i.e. of system plus user simulator) exceeded a threshold, which was 23 interactions for Saplen and 24 for Viajero. Cancelled dialogues were not considered successful and thus they decreased the TC rate.

3.1 Utterance Corpora and Scenarios

We created two separate utterance corpora (for training and test) for each dialogue system ensuring that no training utterances were included in the test corpus. Both corpora include the orthographic transcriptions of the utterances as well as their corresponding reference frames (f_R). For the Saplen system, the training and test corpora contained 2,750 respectively, including product orders, telephone numbers, post codes, addresses, queries, confirmations, error indications, etc. For the Viajero system the corpora contained 2,900 utterances respectively, including travel bookings, telephone numbers, city names, queries, confirmations, error indications, etc.

In order to test the proposed technique and generate dialogues between the dialogue systems and the user simulator, we designed 250 scenarios for each system. The scenario goals were selected by choosing frames at random from the utterance corpora used for testing. In the case of Saplen the frames corresponded to product orders, telephone numbers, post codes and addresses, whereas for Viajero they were concerned with greetings, travel bookings, telephone numbers and queries on travel schedules, price and duration.

3.2 Language Modelling for Speech Recognition

For the Saplen system we employed the two kinds of language model that we used in a previous study [9]: 17 prompt-dependent language models (PDLMs) and one prompt-independent language model (PILM). The PDLMs were word bigrams compiled from training sentences that were representative of each dialogue state. The PILM was a word bigram compiled to recognise any kind of sentence permitted in the application domain regardless of the current system's prompt. For the Viajero system we used the same language modelling, created specifically for this study 16 PDLMs and one PILM. It was interesting for us to test the proposed technique using both kinds of language model because we plan to study ways to let the systems automatically select one or the other as an attempt to adapt their performance to the kind of user (more or less experienced) and the success of system-user interaction (more or less successful).

3.3 Tuning of the Two-Level Speech Recogniser

To identify the best value of the p parameter for the two-level speech recogniser [8] of the Saplen and Viajero systems we tested 21 different values (p = 0, 1, ..., 20). Employing the user simulator we generated one dialogue for each combination of scenario, kind of language model and p value, which makes a total of $250 \times 2 \times 21 = 10,500$ dialogues per system. All misunderstandings and correct understandings obtained by the SLU component of the systems were stored in four models for each system. Two of these stored misunderstandings (one for each language model) and the other two models stored correct understandings (one for each language model). The experimental results let us know that several best values of p must be considered for both systems and the two types of language model. The scores improved as p increased until it reached a threshold which was p = 13 for the PDLMs and p = 10 for the PILM in the case of Saplen, and $p = 12$ for the PDLMs and $p = 10$ for the PILM in the case of Viajero. For values of p greater than these thresholds the scores decreased. Using the optimal values of p, in the case of Saplen WA was 81.13% for the PDLMs and 80.65% for the PILM, whereas SU was 72.66% and 69.71%, respectively. In the case of Viajero, WA was 87.95% for the PDLMs and 84.19% for the PILM, whereas SU was 83.91% and 80.25%, respectively.

3.4 Experiments with the Baseline System

In these experiments the proposed technique was not used and thus the SLU component shown in Fig. 1 was the original SLU component of the dialogue system (Saplen or Viajero). Therefore, the frames received by the system's dialogue manager were not replaced by the frame correction module. Employing the user simulator to interact with each system, we generated 10 dialogues for each scenario and language model using the best value of p in each case. Hence, a total of $10 \times 250 \times 2 = 5,000$ dialogues was generated for each dialogue system. Table 1 sets out the average results obtained from an analysis of these dialogues.

Table 1. Performance of baseline systems (in %)

	PDLMs				PILM			
	SU	**TC**	**WA**	**IR**	**SU**	**TC**	**WA**	**IR**
Saplen	72.66	65.83	81.83	39.4	69.71	63.11	80.65	33.38
Viajero	83.91	78.04	87.95	58.83	80.25	75.21	84.19	52.58

It can be observed that the systems worked slightly better when the PDLMs were used. The reason is that for analysing each response (utterance) provided by the user simulator, the speech recogniser employed a word bigram compiled from training sentences of the appropriate type. In addition, the vocabulary to be considered using the PDLMs was much smaller than when the PILM was used.

3.5 Experiments with the Proposed Technique

In these experiments the Saplen and Viajero systems used the robust speech understanding module. Therefore, the frames generated by the SLU component of the systems were replaced by the frame correction module if they were considered incorrect, which was done before they were used by the dialogue manager of the systems. The two-level speech recogniser was configured to use the best value of p for each language model: p = 13 for the PDLMs and p = 10 for the PILM in the case of Saplen, and p = 12 for the PDLMs and p = 10 for the PILM in the case of Viajero.

Creation of the Initial Correction Model. In accordance with the procedure described in Section 2.1, we created an ICM for each dialogue system to obtain as much knowledge as possible regarding system misunderstandings. For this purpose we used the ICM created in the worst operational case to determine the best value of p (PILM according to Table 1). For the Saplen system this model was called CM1, its size was 22 Megabytes and contained 119,773 tuples. We observed in the experiments that the system had a problem in correctly understanding affirmative and negative confirmations. This error happened because of the Southern Spain accents of most customers of the fast food restaurant that collaborated in the collection of the utterance corpus, who typically omitted the final "s" of plural words in their utterances. We also found system problems in the understanding of product orders and telephone numbers.

For the Viajero system the ICM was called CM2, its size was 20 Megabytes and contained 101,602 tuples. We observed that the system also had difficulties in understanding confirmations, and found a problem with the recognition of some city names due to the acoustic similarity between some names, especially when pronounced by some speakers. It was also noticeable a problem in correctly understanding some telephone numbers due to the confusion in the recognition of at least one digit.

Model Optimisation. In accordance with the procedure described in Section 2.2 we compacted the ICMs for the two dialogue systems to remove repeated tuples, thus obtaining a model of size 19.9 KB for Saplen that contains 164 different tuples, and a model of size 18 KB for Viajero that contains 147 different tuples. In accordance with Section 2.2 we removed the inadequate tuples from the compacted models: 87 in the case of Saplen and 75 in the case of Viajero. As discussed in Section 2.2 we expanded the models to generalise the behaviour of the frame correction module to prompt types not observed in the training. To do this we classified the 43 prompt types that Saplen can generate into 17 classes, and the 52 prompt types that Viajero can generate into 15 classes. As a result we obtained a correction model with 359 tuples (47 KB) for Saplen and another model for Viajero with 320 tuples (45 KB). In order to get experimental results using the robust speech understanding module with these models, we employed again the user simulator and generated 10 dialogues for each scenario and language model, i.e. 10 x 250 x 2 = 5,000 dialogues for each dialogue system. Table 2 sets out the average results obtained from an analysis of these dialogues.

It can be observed that again the system worked better for the PDLMs, which happened because of the reason discussed in the previous section.

Table 2. Performance of systems (in %) employing the proposed technique

	PDLMs				PILM			
	SU	**TC**	**WA**	**IR**	**SU**	**TC**	**WA**	**IR**
Saplen	91.03	91.28	88.9	57.73	89.25	89.36	88.18	53.68
Viajero	98.21	95.90	94.55	74.30	95.18	93.27	91.17	68.21

4 Conclusions and Future Work

A comparison of Tables 1 and 2 shows that the robust speech understanding module improved the performance of the baseline systems. Regarding Saplen, SU increased by 18.37% absolute for the PDLMs (from 72.66% to 91.03%) and by 19.54% absolute for the PILM (from 69.71% to 89.25%). The improvement in terms of SU reflects a remarkable increment in terms of TC, which was 25.45% absolute for the PDLMs and 26.25% absolute for the PILM. The lowest improvement for the two systems was in terms of WA. In the case of Saplen the increment was 7.07% absolute for the PDLMs and 7.53% absolute for the PILM, whereas for Viajero it was 6.60% absolute for the PDLMs and 6.98% absolute for the PILM. The reason for these small increments is that the goal of the proposed technique is to correct frames obtained from the recognised sentences (which obviously affects SU, TC and IR) but not to correct recognised sentences. Directly related to this reason, IR increased notably when the proposed technique was used. In the case of Saplen the increment was 18.33% absolute for the PDLMs and 20.30% absolute for the PILM, whereas for Viajero it was 15.47% absolute for the PDLMs and 15.63% absolute for the PILM. For both systems the proposed technique was slightly more effective for the PILM given that the number of speech recognition errors was somewhat greater for this language model.

Future work includes studying methods to extract more information from the ICM. In the current implementation the technique removes the repeated tuples in the model to reduce it as much as possible in order to avoid any processing delay. However, knowing the number of duplicates of a tuple (T', f_R, f_O) can be important as it provides probabilistic information about how often the dialogue system misunderstands a sentence uttered to answer a prompt type T. Therefore, in our next study we will check whether it is suitable to make frame corrections considering this probabilistic information.

References

1. Higashinaka, R., Sudoh, K., Nakano, M.: Incorporating Discourse Features into Confidence Scoring of Intention Recognition Results in Spoken Dialogue Systems. Speech Communication 48, 417–436 (2006)
2. Jiang, H.: Confidence Measures for Speech Recognition: A survey. Speech Communication 45, 455–470 (2005)
3. Karsenty, K., Botherel, V.: Transparency Strategies to Help Users Handle System Errors. Speech Communication 45, 305–324 (2005)

4. Lemon, O., Gruenstein, A.: Multithreaded Context for Robust Conversational Interfaces: Context-Sensitive Speech Recognition and Interpretation of Corrective Fragments. ACM Transactions on Computer-Human Interaction 11(3), 241–267 (2004)
5. López-Cézar, R., García, R., Díaz, J., Rubio, A.: A Voice Activated Dialogue System for Fast-Food Restaurant Applications. In: Proc. of Eurospeech, pp. 1783–1786 (1997)
6. López-Cózar, R., Rubio, A.J., Garciá, P., Díaz Verdejo, J.E., López-Soler, J.M.: Telephone-Based Service for Bus Travellers Service. In: Proc. of 1st Spanish Meeting on Speech Technologies, Seville, Spain, pp. 181–184 (2000)
7. López-Cózar, R., De la Torre, A., Segura, J.C., Rubio, A.J., Sánchez, V.: Assessment of Spoken Dialogue Systems by means of a New Simulation Technique. Speech Communication 40(3), 387–407 (2003)
8. López-Cózar, R., Callejas, Z.: Two-Level Speech Recognition to Enhance the Performance of Spoken Dialogue Systems. Computer Speech and Language 19, 153–163 (2006)
9. López-Cózar, R., Callejas, Z.: Combining Language Models in the Input Interface of a Spoken Dialogue System. Computer Speech and Language 20, 420–440 (2005)

Linguistic Adaptation in Semi-natural Dialogues: Age Comparison

Marie Nilsenová* and Palesa Nolting

Tilburg Centre for Cognition and Communication
P.O.Box 90153, 5000LE Tilburg, The Netherlands
m.nilsenova@uvt.nl
http://www.tilburguniversity.nl/faculties/humanities/dci/

Abstract. Speaker adaptation in dialogues appears to support not only dialogue coordination, but also language processing, learning and in/out-group manifestation. Presumably, speakers in various stages of their language development might exploit different functions and types of adaptation, but conclusive research in this area has so far been lacking. In the present study, we compare structural, lexical and prosodic adaptation in a semi-natural dialogue across two age groups, in adult-child and adult-adult dyads. The results of our experiments indicate that children take over the structural and lexical forms used by their dialogue partner more frequently than adults. Children also adapt to the pitch of the speaker they interact with more than adult participants. Irrespective of age, we found longer onset latencies following the experimenter's question if the question had a non-canonical (declarative) form compared to a question with a canonical (interrogative) form. This can be seen as a manifestation of a processing advantage typically associated with the long-term effects of adaptation-as-learning.

Keywords: adaptation, alignment, entrainment, prosody, boost effects.

1 Introduction

Starting with the first day of their life, children take over the linguistic patterns they observe in their environment. Arguably, the initial reason for mimicking the speech patterns babies perceive is for them to acquire their caretaker's language. Nevertheless, the same pattern of behavior has been described in older age groups as well. A number of studies have documented that speakers take over each other's choices of lexical expressions, even in situations where as individuals, they might prefer the use of a different lexical item to refer to the object in question [1,2,3]. The effect is also present in human-computer interaction, where it appears to support language processing. As discussed in [4], if users have the idea that the computer software they interact with is low-level and outdated ("basic version 1987" compared to an "advanced version" 2006), they are more likely to adapt their linguistic patterns to the computer in order to facilitate language processing.

[*] The authors would like to thank Marc Swerts for his comments on an earlier version of this manuscript.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 531–538, 2010.

Speakers also tend to take over the syntactic structures of their interlocutors, for example, copying the direct – indirect object word order, the use of a prepositional phrase or of a relative clause [5,6,7,8]. Anecdotal evidence suggests that babies and children adapt their global pitch to that of their primary caretakers [9] or teachers, though an experimental study of global pitch adaptation in children failed to confirm the observations [10]. Other cases of phonetic and phonological adaptation include the pronunciation of vowels and consonants, pitch, accent and speech rate [11,12,13,14,15,16], phonetic imitation in laboratory conditions [17,18,19,20,21] as well as the similarity of neologisms measured in Levenshtein distances [22].

A number of the adaptation processes appear to have a socio-psychological basis in that they serve as a mechanism of group identification. For example, speakers have been shown to adapt their global pitch settings gradually in the course of the interaction to the pitch of the other interlocutor if he or she was considered to be socially more powerful or otherwise more attractive than the speaker [12,14,21].

Interestingly, adaptation on one level of linguistic representation appears to boost the chances of adaptation on another level. In a series of studies based on [6], Branigan [23,4] and others [24,22] showed that if speakers are experimentally steered to use the same verb which they heard in the immediately preceding prime, the likelihood they they will use the same syntactic construction as well increases above chance level. This phenomenon is commonly referred to as a "boost effect" on adaptation and it has been shown to hold also among other levels of representation [22]. The boost effects are modeled in terms of priming percolation in the Interactive Alignment Model (Fig. 1) of Pickering and Garrod [25].

In sum, the process of linguistic adaptation (sometimes also referred to as alignment, mimicking, copying or entrainment) appears to have a number of functions, in that it facilitates long-term learning, language processing support as well as supports important socio-psychological group processes. It is unclear to what extent various age groups exploit these functions and particular levels of linguistic representation are perhaps tied to certain functions (e.g., the prosodic level to group membership processes).

In our experimental study, we addressed the following research questions:

1. Do children adapt (syntactically, lexically and phonetically) more than adult speakers?
2. Is adaptation on a single grammatical level more or less frequent than adaptation on several levels at the same time?
3. Are there any observable long-term adaptation effects (measured with onset latencies)?

2 Experimental Study

2.1 Participants

The participant group consisted of 35 children (4–6y; 16 male) and 58 adults (17–28y; 19 male), all Dutch native speakers. The children were recruited from two elementary schools in Brabant and Limburg; the adult participants were all students from the University of Tilburg who participated in the experiment in exchange for course credit.

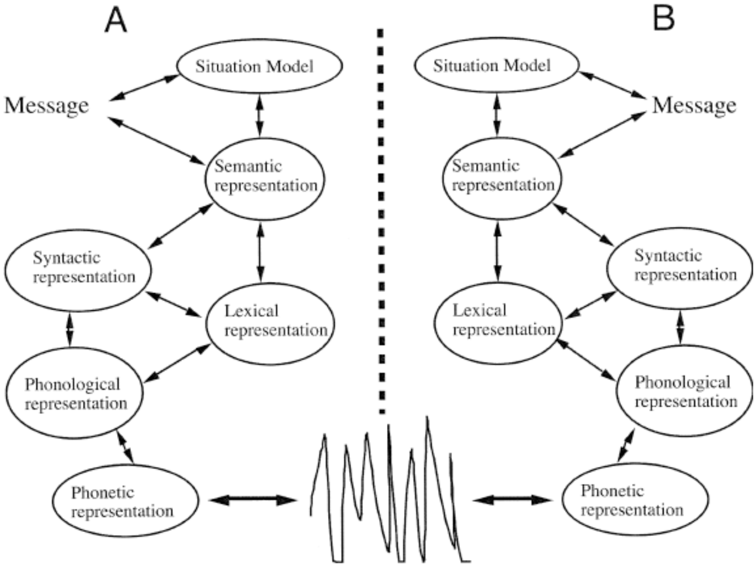


Fig. 1. The Interactive Alignment Model of Pickering & Garrod (2004):177. The model represents the priming effects and percolation processes taking place during a linguistic interaction between speaker A and B. The result, in the ideal case, are two interlocutors who are perfectly 'aligned' to each other's representations.

2.2 Design

The experiment had a mixed within-between design. We systematically manipulated the syntactic structure of the experimenter's questions (declarative *versus* interrogative) and the finite verbs she used in the questions (*be*/*have*). Our dependent variables were the proportion of structural adaptation (taking over the syntactic structure of the experimenter's question), the proportion of lexical adaptation (taking over the choice of the finite verb in the question) and the onset latency of the participant's answer to the experimenter's question. We also measured the level of phonetic adaptation, in particular, the difference between the experimenter's pitch span and the participant's pitch span and the difference between the F0 height of the experimenter's utterance final boundary and the participant's utterance initial boundary.

2.3 Material

To measure adaptation in a semi-natural interactive setting, we developed a dialogue game based on the board game 'Who is it?'. In our experimental game, the players were posing questions to each other to guess who they were. Their identity was determined for them by the other player who chose a card depicting an animal with certain attributes (e.g., a dog with a party hat or a horse with glasses) out of a set with 26 cards. The experimenter started each game by posing a question, e.g., "*Do I have glasses* *on?/I have glasses on?/Am I a horse?/I am a horse?*” and the participant followed by answering the question and posing her own. Prior to the experiment, we pre-tested the pictures on the cards to make sure that the animals and their attributes were recognizable for the younger participant group. For an example of an experimental stimulus (a game card), see Fig. 2.

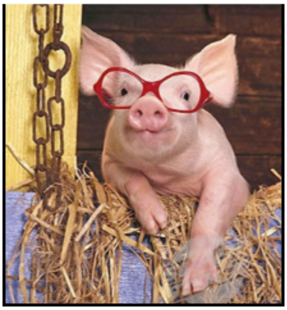


Fig. 2. An example of an experimental stimulus. During the game, the participant and the experimenter were taking turns in guessing which animal they were supposed to be by posing questions about the animal's attributes (“*Do I have glasses?*”) or guessing its identity directly (“*Am I a pig?*”).

2.4 Procedure

The experiment took place in a quiet classroom. To prevent any effects of nonverbal behavior on the outcome of the experiment, the experimenter and the participant were separated by a screen that also served as a pinboard for the the game cards. The experimental session consisted of an introductory part with an explanation, a trial game and a series of real games. In order to prevent the participants from guessing the purpose of the experiment, they were told that the experimenter was testing a prototype for a new computer game.

The dialogue during the game was partly structured in that the experimenter controlled for a systematic variation in the two different types of syntactic structures and two different types of finite verbs (+ a number of fillers with other verbs), but did not control for the length of an individual game. Depending on how the games developed, a participant would play between 2 – 8 real games which could vary in duration between 10–20 minutes for the adult-child dyads (M = 14.1 min, SD = 3.8) and 8–14 minutes for the adult-adult dyads (M = 10.3 min, SD = 1.4).

The dialogues were recorded on a laptop with a built-in microphone, using the Audacity software, v. 1.2.5., in mono at a 44,100 Hz sampling rate, and subsequently analyzed with the Praat software, v.5.1.03. The prosodic parameters needed for the analysis (F0 span, F0 at utterance boundary and pause duration) were measured directly on the Praat pitch track and the values were perceptually validated to correct for possible octave jumps and noise effects.

2.5 Results

In order to answer the research questions stated above, we first compared the proportions of syntactic structures that participants took over from the experimenter. We categorized as *canonical* the questions with an interrogative word order, since they were the less marked syntactic choice in the context of the game.[1] We categorized as *non-canonical* the questions with the declarative word order.[2] A mixed within-between analysis of variance showed that there was a significant main effect of the type of question (canonical vs. non-canonical) on the proportion of adapted structures, F(1,89) = 264.01, p<.05, Eta-squared = .75, in that the experimental participants were more likely to take over the canonical structures compared to the non-canonical ones ($M_{canonical}$ = 47.5, *SD* = 1.4 and $M_{non-canonical}$ = 11.6, *SD* = 1.6). There was no significant main effect of age but the interaction effect between type of question and age was significant, F(1,89) = 11.3, p<.05, Eta-squared = .11, in that children were more likely to take over the non-canonical structures than adult participants (see Table 1). Note that in the case of canonical structures, it is not clear whether speakers are adapting to their use because of the choices made by their interlocutor or because it would be their default choice anyway. Non-canonical instances of adaptation are thus a more reliable indicator that the speaker is, in fact, influenced by what she hears (the prime).

Table 1. The interaction effect between the proportion of syntactic adaptation and age, including the means of percentages for the canonical and non-canonical questions (total number of questions = 100%)

	Canonical Question		Non-canonical Question		
Group	*M*	*SD*	*M*	*SD*	Eta-squared
Child	44.4	2.1	15.8	2.4	.11*
Adult	50.7	1.8	7.3	2.0	

*p<.05

With respect to lexical adaptation, there was no significant main effect of the verb type (*be* versus *have*) but there was a significant main effect of age, F(1,89) = 19.99, p<.05, Eta-squared = .18. Children took over the verb used by the experimenter more frequently than adults ($M_{children}$ = 40.1, *SD* = 1.5, M_{adults} = 31.5, *SD* = 1.2). There was also a significant interaction effect between verb type and age. As shown in Table 2, children were more likely to adapt to the use of the verb *have* than adults and less likely to take over the verb *be*.

As for the percolation (boost) effects between the syntactic and the lexical level of representation, we found that speakers were more likely to use the same syntactic structure *and* the verb as the experimenter (*M* = 46.3, *SD* = 1.8), compared to just using

[1] Declarative questions are typically used to elicit confirmation, whereas the questions in the game were used to enquire about a new issue each time.

[2] Both the canonical and non-canonical questions were always pronounced with a rising intonation.

Table 2. The interaction effect between the proportion of lexical adaptation and age, including the means of percentages for the two verbs used in the questions (total number of questions = 100%)

	verb *BE*		verb *HAVE*		
Group	*M*	*SD*	*M*	*SD*	Eta-squared
Child	17.2	2.9	63.1	3.5	.56*
Adult	47.6	2.4	15.3	2.9	

*p<.05

the same syntactic structure (M = 12.8, SD = 1.0) or only the same verb (M = 25.3, SD = 1.5) or not adapting at all (M = 9.4, SD = .8), F(1,89) = 44.68, P<.05, Eta-squared = .33. There was no significant main effect of age.

We also explored the relation between the pitch span of the experimenter and the participants. We found that the pitch span of the children participating in the experiment differed less from the pitch span of the experimenter than the pitch span of the adult participants, F(1, 7) = 20.8, p < .05, Eta-squared = .75 ($M_{children}$ = 185.8 Hz, SD = 6.2, M_{adults} = 251.2, SD = 11.1). We found no other effects regarding pitch span and no interaction between the syntactic and lexical adaptation and the adaptation in pitch span, respectively. The comparison of the F0 at the utterance boundaries (experimenter's final F0 and the participants' initial F0) gave a comparable result, in that the F0 height of the participating children differed less from the experimenter's value than the F0 of the adult participants, F(1, 7) = 92.77, p< .05, Eta-squared = .93 ($M_{children}$ = 134.3 Hz, SD = 4.2, M_{adults} = 248.9, SD = 13.0).

Finally, we examined the effect of the question type on the onset latencies for its answer (measured up to the beginning of a filled pause, if there was one). We predicted that from the perspective of long-term adaptation effects, if the experimenter used a canonical question, the participant would be able to process it more quickly than a non-canonical question. Our data confirmed the prediction, in that canonical questions resulted in shorter onset latencies (M = 80 ms, SD = 17) than non-canonical questions (M = 138 ms, SD = 29), F(1, 87) = 4.07, p<.05, Eta-squared = .05.[3]

3 Conclusion and Discussion

In our experiment, we found that compared to adults, children were more likely to take over both the syntactic structure (an effect visible if we focus on non-canonical structures) and the finite verb used by the experimenter. Similarly, they appeared to adapt their pitch span and utterance initial F0 more than the adult participants. These results suggest that disregarding the type of linguistic representation and the possibly different functions of adaptation, children in this particular age group, i.e., preschoolers, are highly sensitive to the linguistic input they receive and take it over. Interestingly,

[3] We also observed that adult participants gave the answer significantly faster than the participating children, which explains why on average, they played more game rounds in the allotted time (see Procedure).

though, compared to adults, they were reluctant to take over identity-changing questions ("*Am I a dog?*"), which may be due to the semantics of the question involving an identity change. Finally, we also observed that the use of canonical questions results in shorter onset latencies. This can be ascribed to a processing advantage related to adaptation.

To our knowledge, the present study is the first attempt to explore systematically different types of adaptation processes in different age groups. Apart from testing more representational levels, it still remains to be seen if the results can be generalized to other interactive situations and other dyads, involving, e.g., a male experimenter.

References

1. Garrod, S., Anderson, A.: Saying What You Mean in Dialogue: A Study in Conceptual and Semantic Coordination. Cognition 27, 181–218 (1987)
2. Metzing, C., Brennan, S.E.: When Conceptual Pacts Are Broken: Partner-Specific Effects on the Comprehension of Referring Expressions. Journal of Memory and Language 49, 237–246 (2003)
3. Garrod, S., Doherty, G.: Conversation, Co-Ordination, and Convention: An Empirical Investigation of How Groups Establish Linguistic Conventions. Cognition 53, 181–215 (1994)
4. Branigan, H.P., Pickering, M.J., Pearson, J., McLean, J.F.: Linguistic Alignment Between People and Computers. Journal of Pragmatics (2008)
5. Levelt, W.J.M., Kelter, S.: Surface Form and Memory in Question Answering. Cognitive Psychology 14, 78–106 (1982)
6. Bock, K.: SyntactIC Persistence in Language Production. Cognitive Psychology 18, 355–387 (1986)
7. Pickering, M.J., Branigan, H.P.: Syntactic Priming in Language Production. Trends in Cognitive Science 3, 136–141 (1999)
8. Gries, S.T.: Syntactic Priming: A Corpus-Based Approach. Journal of Psycholinguistic Research 34, 365–399 (2005)
9. Liberman, P.: Intonation, Perception, and Language. The MIT Press, Cambridge (1967)
10. Nilsenová, M., Swerts, M.G.J., Houtepen, V., Dittrich, H.: Pitch Adaptation in Different Age Groups: Boundary Tones versus Global Pitch. In: Proceedings of Interspeech, Brighton, September 6-10 (2009)
11. Natale, M.: Converge of Mean Vocal Intensity in Dyadic Communication as a Function of Social Desirability. Journal of Personality and Social Psychology 32, 790–804 (1975)
12. Gregory, S.W., Hoyt, B.R.: Conversation Partner Mutual Adaptation as Demonstrated by Fourier Series Analysis. Journal of Psycholinguistic Research 11, 35–46 (1982)
13. Giles, H., Coupland, N., Coupland, J.: Accommodation Theory: Communication, Context, and Consequence. In: Giles, H., Coupland, J., Coupland, N. (eds.) Contexts of Accommodation, pp. 1–68 (1991)
14. Gregory, S.W., Gallagher, T.J.: Spectral Analysis of Candidates' Nonverbal Communication: Predicting U.S. Presidential Election Outcomes. Social Psychology Quarterly 49, 237–246 (2002)
15. Pardo, J.S.: On Phonetic Convergence during Conversational Interaction. Journal of the Acoustical Society of America 119, 2382–2393 (2006)
16. Delvaux, V., Soquet, A.: The Influence of Ambient Speech on Adult Speech Productions through Unintentional Imitation. Phonetica 64, 145–173 (2007)

17. Goldinger, S.D.: Perception and Production in an Episodic Lexicon. In: Johnson, K., Mullennix, J. (eds.) Talker Variability in Speech Processing, pp. 33–66. Academic Press, San Diego (1997)
18. Goldinger, S.D.: Echoes of Echoes? An Episodic Theory of Lexical Access. Psychological Review 105, 251–279 (1998)
19. Goldinger, S.D., Azuma, T.: Puzzle-Solving Science: the Quixotic Quest for Units in Speech Perception. Journal of Phonetics 31, 305–320 (2003)
20. Namy, L.L., Nygaard, L.C., Sauterberg, D.: Gender Differences in Vocal Accommodation: the Role of Perception. Journal of Language and Social Psychology 21, 422–432 (2002)
21. Babel, M.E.: Phonetic and Social Selectivity in Speech Accommodation. Ph.D. dissertation, University of California, Berkeley (2009)
22. Nilsenová, M., van Amelsvoort, M.A.A.: Syntactic Boost Effects on Phonological Priming in Dutch. Ms., University of Tilburg (2010)
23. Branigan, H.P., Pickering, M.J., Cleland, A.A.: Syntactic Coordination in Dialogue. Cognition 75, B13–B25 (2007)
24. Hartsuiker, R.J., Bernolet, S., Schoonbaert, S., Speybroeck, S., Vanderelst, D.: Syntactic Priming Persists while the Lexical Boost Decays: Evidence from Written and Spoken Dialogue. Journal of Memory and Language 58, 214–238 (2008)
25. Pickering, M., Garrod, S.: Toward a Mechanistic Psychology of Dialogue. Behavioral and Brain Sciences 27, 169–226 (2004)

Automatic Speech Recognition Based on Multiple Level Units in Spoken Dialogue System for In-Vehicle Appliances

Masafumi Nishida[1], Yasuo Horiuchi[2], Shingo Kuroiwa[2], and Akira Ichikawa[3]

[1] Faculty of Science and Engineering, Doshisha University, Kyoto, Japan
[2] Graduate School of Advanced Integration Science, Chiba University, Chiba, Japan
[3] Faculty of Human Sciences, Waseda University, Saitama, Japan
mnishida@mail.doshisha.ac.jp

Abstract. The purpose of our study is to develop a spoken dialogue system for in-vehicle appliances. Such a multi-domain dialogue system should be capable of reacting to a change of the topic, recognizing fast and accurately separating words as well as whole sentences. We propose a novel recognition method by integrating a sentence, partial words, and phonemes. The degree of confidence is determined by the degree to which recognition results match on these three levels. We conducted speech recognition experiments for in-vehicle appliances. In the case of sentence units, the recognition accuracy was 96.2% by the proposed method and 92.9% by the conventional word bigram. As for word units, recognition accuracy of the proposed method was 86.2% while that of whole word recognition was 75.1%. Therefore, we concluded that our method can be effectively applied in spoken dialogue systems for in-vehicle appliances.

Keywords: spoken dialogue system, in-vehicle appliances, automatic speech recognition, multiple recognition units.

1 Introduction

We focus on the functions of in-vehicle appliances, because the speech interface seems to be of particular importance when a person is driving a car and needs to be “eyes and hands free”. The purpose of our study is to develop a multi-domain spoken dialogue system that controls the various in-vehicle appliances, such as air-conditioners, audio-players, navigation systems.

A multi-domain dialogue system should be capable of reacting to a change of topic in the utterance of users. The method using a two-stage model has been proposed: in the first step, a speech utterance is recognized by language model with the combination of whole domain, and in the second step, a domain is selected from the results obtained in the first step and utterances are recognized by the language model of the selected domain [1]. If we have large vocabulary as candidates, vast quantity of data is needed for making the language model. It increases the risk of generating unavailable recognition result for the system.

To avoid that risk, we construct all functions of the in-vehicle appliances by a tree structure. We defined the dialogue situation based on dialogue act categorized by

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 539–546, 2010.

dialogue function. In each dialogue situation, predicted utterances are registrated based on sentence units. The system recognizes the speech utterance based on sentence units and judges the dialogue situation based on a sentence which likelihood is maximum. By this method, we can avoid the risk of the generating unavailable recognition result for the system and the dialogue is controlled without grammatical rules.

Conventional recognition methods mainly deal with word units and use the difference of likelihood between acoustic model and sub-word model [2,3,4]. However, the acoustic information seems to be insufficient because it is important for spoken dialogue to reduce the competitive candidates down for finding accurate answer immediately. The ROVER method determines the recognition result by the rule of majority from the results through many recognition systems [5,6]. However, it is not applicable for in-vehicle appliances because minimum combination of recognition systems is required from the view point of processing cost.

We propose a novel recognition method in order to address these problems. We first prepare two parallel recognition systems including phonemes and partial word recognition. The system automatically reduces the candidates down to sentence and word units based on the degree of reliability which is determined by the degree of coincidence of these recognition results. In this study, we examined the efficiency of our method by recognition experiments for four domains, namely, an air-conditioner, audio-player, radio, and navigation system.

Step ID: U1

Category for utterance:
Request for unknown info.

Predicted sentences:
<F> +mokutekichi <PN> desu. (<F> the goal is <PN>.)
<F> <PN>made +onegai shimasu. (<F> please go to <PN>.)
<F> <PN>ni +ikitaindesukedo. (<F> I want to go to <PN>.)

Fillers: <F>
Eeto, Ah, Eh. (None)

Proper names: <PN>
Keiyo Ginko(Keiyo Bank), Seibu Hyakkaten(Seibu Department Store), Chiba Daigaku(Chiba University), Tokyo Eki(Tokyo Station) ... etc.

Fig. 1. Example of predicted sentence

2 Spoken Dialogue System Based on Prediction for In-Vehicle Appliances

We defined several functions of the multi-domains such as temperature, sound volume, broadcast for the radio, play and stop for the audio-player, destination setting. We first define all functions of the in-vehicle appliances in a tree data structure and registered some possible utterance for each function. We set a state transition model corresponding to the function and control the dialogue by the model. In the state transition model, we defined dialogue states from the tag of utterance unit such as “request for information”, “confirmation” and “affirmation/ negation”. In the step of utterance of system, some states are registered: the kinds of utterance, the system utterance at the current state and the predictable step of a user's utterance. The system transits the current several predicted states to the predicted state that has a sentence with maximum likelihood from the recognition result on the use's utterance. On the other hand, in the step of user's utterance, the kinds of utterance, the user's predictable utterance at the current state, and the next step of utterance of system are registered.

The system recognizes the candidates of predicted sentence that are registered corresponding to each function by using one path search. Fig.1 shows the example of predicted sentence. The predicted sentences from user's utterance are formed as slots. The total number of predicted sentences is 11,020: 1,276 for the air-conditioner, 1,744 for the radio, 1,696 for the music and 6,304 for the navigation system. For example, there are predicted sentences such as "Set the temperature to 25 degrees!", "Turn up the air volume!", "Turn up the sound volume!", "Change to Tokyo FM!", "Play music!" and "Go to the convenience-stores!".

3 Reduction of Candidates by Multiple Recognition Units

Speech recognition accuracy tends to decrease when it deals with major lexicons. Therefore, if the system tries to recognize the candidates of all domains, it needs to ask a user to confirm repeatedly his/her answer. One of the factor of misrecognitions is that the system needs to recognize more objects. However, it is generally difficult to optimize the beam width corresponding to the number of objects in advance. We propose a method that the system reduces the number of candidates as sentence and word units based on the reliability degree of results. Fig. 2 shows the process by the proposed method. First, the system reduces the number of candidate sentences by DP (Dynamic Programming)-matching between the phoneme sequences obtained from

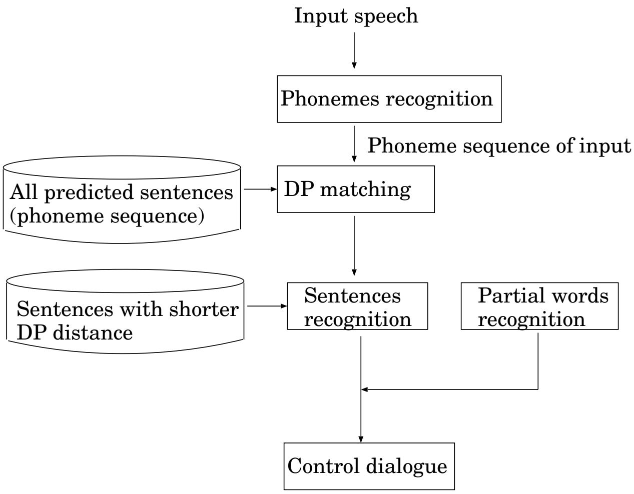

Input speech
Phonemes recognition
Phoneme sequence of input
All predicted sentences (phoneme sequence)
DP matching
Sentences with shorter DP distance
Sentences recognition
Partial words recognition
Control dialogue

Fig. 2. Process by proposed method

phoneme recognition and those of all candidates for the predicted sentence. At this only one stage of process, therefore, the system can process the recognition and the estimation for both the domain and the function.

Secondly, we describe the method for reducing the number of candidates at the word level. The situation when the candidate-words appear frequently seems to be the goal setting on the navigation system. In order to achieve efficient recognition of the destination, we designed a system that narrows down the number of candidates by performing both sentence and the partial word-recognition. In the recognition system at the sentence level, proper names such as “Chiba Daigaku (Chiba University)” and “Keiyo Ginko (Keiyo Bank)” are registered in the dictionary. However, if the system relies only on this device to perform recognition, it may repeatedly ask the driver to confirm the answer, since it needs to verify the whole word even if it recognizes parts of the word correctly. We, therefore, included also a partial word recognition system, in which the proper names are divided into two slots: the name slot and attribution slot. The names are registered such as “Chiba” and “Keiyo” in the anterior slot, while the attribution such as “Daigaku (University)” and “Ginko (Bank)” is registered in the posterior slot. The system of partial-word recognition outputs the result by combining results from both the name and the attribution slots. In this way, the system is capable of processing unfamiliar words by combining the name and the attribution.

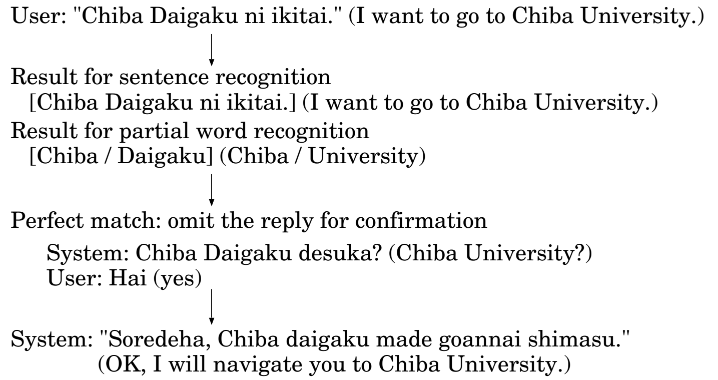
If we categorize the results from the two recognition systems for the predicted word and the partial word, there are three patterns such as perfect, partial, and imperfect match. The system uses these three patterns as a measure of reliability and narrows down the number of candidates. Then it searches again among the reduced candidates and finally decides on the answer. Examples of the three patterns are illustrated in Fig. 3, 4 and 5 respectively.

As shown in Fig. 3, when the system has a perfect match result, it decides that the reliability degree is high enough to select the candidate as the final result, and omits the reply to confirm the result. As shown in Fig. 4, when the system has a partial match, it judges that the reliability degree is high enough to reduce the number of candidates to those having part of the word. The system then searches again among the reduced candidates. As shown in Fig. 5, when the system has imperfect match, it judges that there may be correct result in either the recognition system for predicted sentence or that for partial word. The system extends the number of candidates that have names of four words including the anterior partial words and the posterior ones in the results of recognition on predicted sentence, and the anterior and posterior ones in the results of recognition on partial words.

4 Experiments

4.1 Reduction of Candidates by Sentence and Phoneme Units

We conducted recognition experiments in order to demonstrate the efficiency of our method. We used the julius-3.1 [7] speech decoder. We set the number of states in the acoustic model at 3,000, mixtures distribution at 64, and we used a gender-independent PTM triphone. The phoneme recognition was performed by using the PTM triphone.



User: "Chiba Daigaku ni ikitai." (I want to go to Chiba University.)

Result for sentence recognition
[Chiba Daigaku ni ikitai.] (I want to go to Chiba University.)

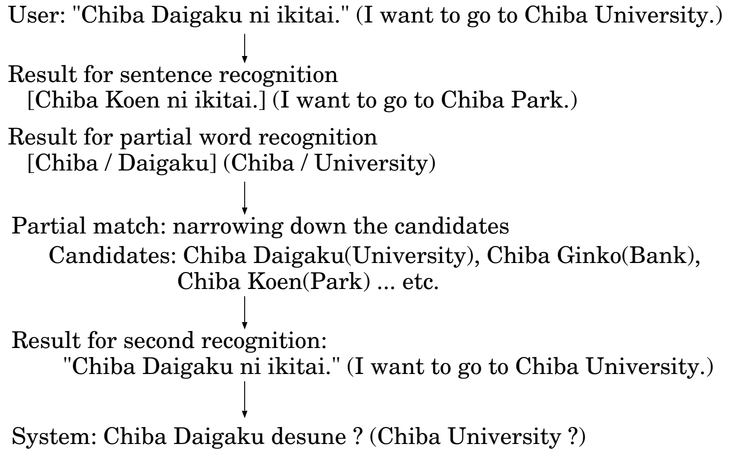
Result for partial word recognition
[Chiba / Daigaku] (Chiba / University)

Perfect match: omit the reply for confirmation

System: Chiba Daigaku desuka? (Chiba University?)
User: Hai (yes)

System: "Soredeha, Chiba daigaku made goannai shimasu."
(OK, I will navigate you to Chiba University.)

Fig. 3. Example of perfect match



User: "Chiba Daigaku ni ikitai." (I want to go to Chiba University.)

Result for sentence recognition
[Chiba Koen ni ikitai.] (I want to go to Chiba Park.)

Result for partial word recognition
[Chiba / Daigaku] (Chiba / University)

Partial match: narrowing down the candidates
Candidates: Chiba Daigaku(University), Chiba Ginko(Bank),
Chiba Koen(Park) ... etc.

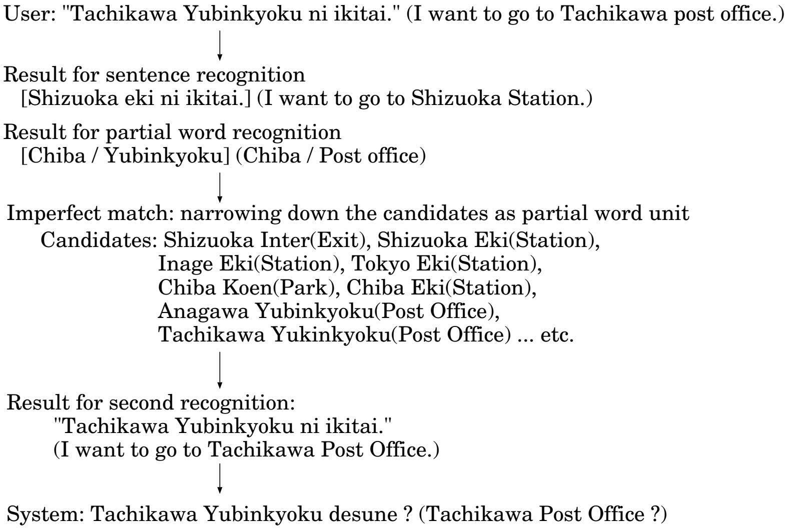
Result for second recognition:
"Chiba Daigaku ni ikitai." (I want to go to Chiba University.)

System: Chiba Daigaku desune ? (Chiba University ?)

Fig. 4. Example of partial match

In order to achieve real-time recognition in the system, we set the beam width on the decoder at optimal value.

Initially, we conducted an experiment to compare our method with the conventional method based on word bigram. The word bigram was trained using the predicted sentences (11,020 sentences). We used 500 utterances from 5 speakers in the recognition experiment. Out of vocabulary and sentence were not included in the experimental data. The highest accuracy rate of our method was 96.2% obtained when it reduced the number of candidates to the best 100 by DP-matching, while that of the word bigram-based method was 92.9%. These results imply that the conventional method possibly generates unacceptable utterances for the system, while the proposed method avoids such risk by performing recognition on the sentence unit level.



User: "Tachikawa Yubinkyoku ni ikitai." (I want to go to Tachikawa post office.)

Result for sentence recognition
[Shizuoka eki ni ikitai.] (I want to go to Shizuoka Station.)

Result for partial word recognition
[Chiba / Yubinkyoku] (Chiba / Post office)

Imperfect match: narrowing down the candidates as partial word unit
Candidates: Shizuoka Inter(Exit), Shizuoka Eki(Station),
Inage Eki(Station), Tokyo Eki(Station),
Chiba Koen(Park), Chiba Eki(Station),
Anagawa Yubinkyoku(Post Office),
Tachikawa Yukinkyoku(Post Office) ... etc.

Result for second recognition:
"Tachikawa Yubinkyoku ni ikitai."
(I want to go to Tachikawa Post Office.)

System: Tachikawa Yubinkyoku desune ? (Tachikawa Post Office ?)

Fig. 5. Example of imperfect match

Furthermore, we analyzed the efficiency of our method by using a total of 2,000 utterances from 10 speakers. In Table 1, "All sentences" refers to the case when the candidates were all sentences for all domains, "Selected domain" refers to the case when a domain was selected only when judged to be 100% correct, "Proposed method" refers to the case when the system narrowed down the candidates to the 100 closest sentences before performing recognition. Clearly, the proposed method achieves approximately 3% higher accuracy rate thus allowing us to confirm its validity.

Table 1. Accuracy for speech recognition on sentences

All sentences	Selected domain	Proposed method
87.9%	93.7%	96.3%

4.2 Reduction of Candidates by Sentence and Word Units

We evaluated the method for reducing the number of candidates by both predicted sentences and partial word recognitions. In the dictionary, 250 formal names were registered (e.g. Chiba Daigaku (Chiba University), Keiyo Ginko (Keiyo Bank)). In the partial word recognition, 77 words were registered for the anterior slot (e.g. Chiba, Keiyo) and 66 words were registered for the posterior slot (e.g. Daigaku, Ginko). We asked 10 speakers to speak names in the dictionary, such like "Chiba Daigaku". We used 3,978 utterances in the experiment. Out of vocabulary and sentence were not included in the experimental data.

Table 2 shows the percentage for matching between two recognitions. The correct answer in perfect match means that all proper names were matched among the user's voice and the two recognition results (e.i. the user spoke "Chiba Diagaku Desu (Chiba University)", the result from predicted sentence recognition was "Chiba Daigaku Desu", and the result from partial word was "Chiba, Daigaku"). The partial match means that two names were matched between user's speech and matched result in the two recognitions (i.e. the user spoke "Chiba Diagaku Desu (Chiba University)", the result from predicted sentence recognition was "Chiba Koen (Park) Desu", and the result from partial word was "Chiba, Daigaku").

Table 2. Accuracy rate by parallel recognition of sentence and partial word units

	Perfect match	Partial match
Degree of coincidence	72.8% (2,897/3,978)	9.7% (384/3,978)
Accuracy rate	99.2% (2,874/2,897)	44.0% (169/384)

Table 3. Result for second recognition with same utterances

	Sentence recognition	Proposed method
Partial match	20.1% (77/384)	34.6% (133/384)
Imperfect match	5.0% (35/697)	60.5% (422/697)

From these results it seems that the reliability of recognition result increased by two recognitions. This recognition method seems to lead the process to the next step without extra reply for confirming the user's answer. The partial match seems that the reliability of the result is higher than the case when two systems output different result even if the result is wrong. Table 3 shows the recognition results for the partial and imperfect matchs. In the case of predicted sentence, the recognition accuracy is 20.1%. When the system narrows down the candidates to the words which are partially matched, the recognition accuracy is increased to 34.6%. The result shows the possibility to decide

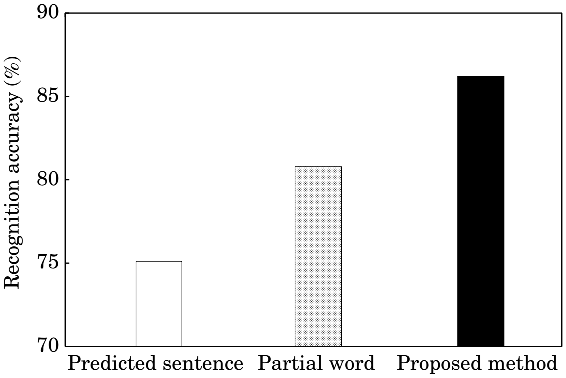


Fig. 6. Accuracy for each method

the recognition result immediately, by confirming the final result after narrowing down the candidates based on partial words from two recognition systems.

Fig. 6 shows the accuracies for the recognition results between the method with each one recognition system and the proposed method. The recognition accuracies were 75.1% for only sentence recognition, 80.8% for only partial word recognition and 86.2% for proposed method. We confirmed that the proposed method can obtain recognition results faster and more accurately compared to conventional methods.

5 Conclusions

In this study, we proposed a novel recognition method for a multi-domain spoken dialogue with in-vehicle appliances. The method was used the degree of reliability by the degree of coincidence in three recognition systems for phonemes, partial words and sentences. As the results of experiments, the accuracy of proposed method for sentences was 96.2%, while that of the conventional method using word bigram was 92.9%. Additionally, the accuracy of the proposed method for words was 86.2% when it relied on partial word recognition, while that of the conventional method relying on whole word recognition was 75.1%. These results confirm the validity of the proposed method which uses as confidence measures based on multiple recognition units.

As a future work, we will study to unify the proposed method and other additional informations. Moreover, we will treat out of vocabulary and sentence in the system and conduct experiments in car environments.

Acknowledgments. This work was supported by Grant-in-Aid for Young Scientists (B), 21700184.

References

1. Lane, I. R., Kawahara, T., Matsui, T.: Language Model Switching Based on Topic Detection For Dialog Speech Recognition. In: Proc. ICASSP, pp. 616–619 (2003)
2. Hazen, T.J., Seneff, S., Polifroni, J.: Recognition Confidence Scoring and Its Use in Speech Understanding Systems. Computer Speech and Language 16, 49–67 (2002)
3. Hirschberg, J., Litman, D., Swerts, M.: Prosodic and Other Cues to Speech Recognition Failures. Speech Communication 43, 155–175 (2004)
4. Raymond, C., Bechet, F., Mori, R. D., Damnati, G., Esteve, Y.: Automatic Learning of Interpretation Strategies for Spoken Dialogue Systems. In: Proc. ICASSP, vol. 1, pp. 425–428 (2004)
5. Fiscus, J. G.: A Post-Processing System to Yield Reduced Error Word Rates: Recognizer Output Voting Error Reduction. In: Proc. ASRU, pp. 347–354 (1997)
6. Schwenk, H., Gauvain, J. L.: Combining Multiple Speech Recognizers Using Voting and Language Model Information. In: Proc. ICSLP, vol. 2, pp. 915–918 (2000)
7. Lee, A., Kawahara, T., Shikano, K.: Julius – an Open Source Real-Time Large Vocabulary Recognition Engine. In: Proc. EUROSPEECH, pp. 1691–1694 (2001)

Dialogue System Based on EDECÁN Architecture

Javier Mikel Olaso and María Inés Torres

Universidad del País Vasco
{javiermikel.olaso,manes.torres}@ehu.es

Abstract. Interactive and multimodal interfaces have been proved of help in human-machine interactive systems such as dialogue systems. Facial animation, specifically lips motion, helps to make speech comprehensible and dialogue turns intuitive. The dialogue system under consideration consists of a stand that allows to get current and past news published on the Internet by several newspapers and sites, and also to get information about the weather, initially of Spanish cities, although it can be easily extended to other cities around the world. The final goal is to provide with valuable information and entertainment to people queuing or just passing around. The system aims, as well, at disabled people thanks to the different multi-modal input/outputs taken into consideration. In this work are described the diferent modules that are part of the dialogue system. These modules where developed under EDECÁN architecture specifications.

Keywords: Dialogue systems, EDECÁN architecture.

1 Introduction

Interactive and multimodal interfaces have been proved of help in human-machine interactive systems such as dialogue systems. Facial animation, specifically lips motion, helps to make speech comprehensible and dialogue turns intuitive. The dialogue system under consideration consists of a stand that allows to get current and past news published on the Internet by several newspapers and sites, and also to get information about the weather, initially of Spanish cities, although it can be easily extended to other cities around the world. The final goal is to provide with valuable information and entertainment to people queuing or just passing around. The system aims, as well, at disabled people thanks to the different multi-modal input/outputs taken into consideration. Apart from keyboard, mouse and screen, the stand counts on a web-cam, a microphone, and loudspeakers.

A typical dialogue would proceed as follows: the screen, divided in four pieces, in one of the pieces is permanently showing information, that periodically changes randomly from one site to the other; in another piece there is an animated character in a *waiting* friendly attitude, blinkering but without much movement. This is meant to catch user's attention. By the time a user is close to the stand, a face detection device detects the prospective user and starts the character that encourages the user to look for information. The interactions carried out by the character are performed by means of both text and speech. Next, the user is supposed to interact through either the keyboard, the mouse or simply by talking to the character using natural language. The required

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 547–551, 2010.

information is requested through the net and displayed in the screen. Additionally, the information can be uttered by the character. The system puts an end to the dialogue by either a specific request from the user or when the face detection device keeps on without detecting a face for a pre-stablished period of time.

In this work are described the diferent modules that are part of the dialogue system. These modules where developed under EDECÁN [1,2] architecture specifications[1]. This architecture was developed with the main goal of communicate any kind of services through TCP/IP protocols, specially those services needed in the implementation of a dialogue system, under Linux and Windows operating systems. In [2] can be found a detailed description of the architecture.

In next section the different modules taking part in the dialogue system are described. Finally, conclusions and open issues to be tackled are proposed.

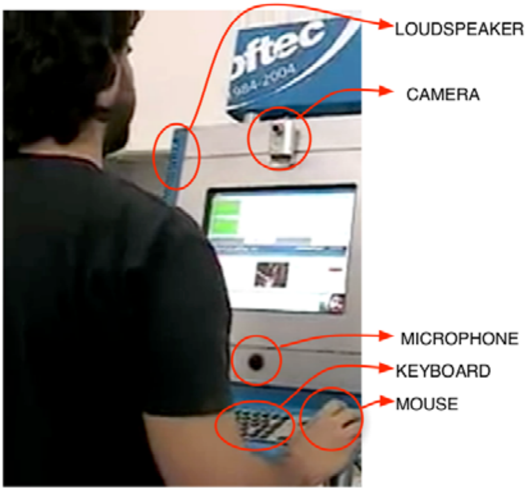


Fig. 1. Final system

2 Architecture of the Multimodal Dialogue System

The dialogue system under consideration is an informatic system whose inputs are speech phrases and whose outputs are synthetic speech phrases. The main goal of the dialogue system is the emulation of human intelligent behavior in a specific task. Now are described the basic system modules and those reponsible of information visualization. In Figure 2 are schematically shown the main modules of the system.

- Face recognition module: As for user-presence detection device a face recognition system is used. It has been implemented using OpenCV [5]. OpenCV is a free

[1] Project supported by the Ministry of Science and Innovation, 2005–2008, led by the University of Zaragoza and with participation of University of the Basque Country, Polytechnique University of Valencia and Polytechnique University of Madrid. [3].

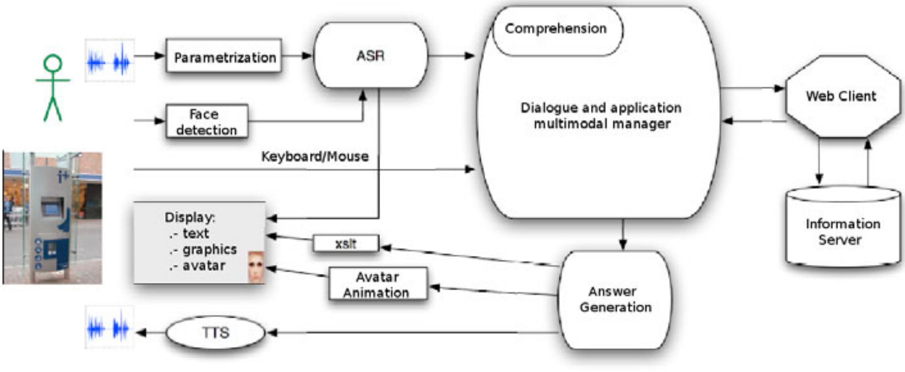


Fig. 2. Multimodal dialogue system: Basic architecture

library of programming functions mainly aimed at real time computer vision. This software allows to adjust the enable-disable periods required to start/quit the speech dialogue system. In general terms, the system is started a configurable amount of time (3 seconds in this case) after being recognised a face.

- Speech recognition module (ASR): Contains a preprocess submodule that gets the acoustic signal provided by the user and transforms it obtaining an observation sequence, in a space represented by: energy, cepstral coefficients and first and second derivatives of cepstral coefficients. This observation sequence, then, serves as input for the automatic recognition system which transform it in the most likely word sequence, acording to the task's language model and the acoustic models for the specific language, castilian in this prototype. The speech recognition system integrated in this module was completely developed by Pattern Recognition & Language Technologies group at the University of the Basque Country. It is a medium size vocabulary recognizer with Hidden Markov Models to implement acoustic models and stocastic finite state machines to implement language models.
- Understanding module: This module extracts the meaning from the word sequence given by the speech recognition system. That is, it provides a semantic representation of the input data (word sequence).
- Dialogue manager: It considers the semantic representation of the user's request, the history of the dialogue, the information that is available just at that moment and determines the strategy of the dialogue. The goal is to get from the user the minimum amount of queues to carry out a task. Also is responsible of answers generation and visualization modules activation. In the system described in this article understanding and dialogue manager modules where implemented as single module and more complicated tasks would need to separate them in diferent modules.
- webclient module: The information that the system finally shows on screen to the user is obtained from different sites in the Internet. Therefore, it's necesary a system able of accessing and obtaining the information required by the user. This module is activated once the dialogue manager has collected all the necessary information from the user.

- xslt module: Information obtained by the webclient module is not in an apropriate format to be shown to the user. Due to the fact that the visualization module uses HTML format to show the information on screen and the data obtained from the Internet are in XML format, it's necessary a previous conversion between the two formats.
- Visualization module: This module is dedicated to show information on screen. The information can be of three types: text, graphics and an avatar. Text and graphics are used to show information to the user. Also it shows other information relative to the state of the system:
  - Information relative to the state of the speech recognizer: With this information can be known if the recognizer is in audio acquiring state or in recognition state.
  - Recognized text: On screen is shown the text asociated to the audio signal provided by the user, that is result of the speech recognition process.
  - Face detection system state: Also is shown to the user an image of his own face. Over this image a circle is drawn when the system detects user's presence.
- Avatar animation: An avatar was generated in order to provide an enhanced interaction among computer and user. The avatar was built by making use of the *iclone* tool of the free version of *Reallusion* [6]. This tool provides several avatar models that can be changed and adapted to the programmer's needs and preferences. The avatar was also provided with different movements in order to achieve a more realistic interaction. When it comes to animating a character's lips, synchronization with speech-to-text (TTS) device represents a key issue to bear in mind. For a correct synchronization between the audio and the avatar movements it's necessary to know the phoneme chain to be uttered by the synthesiser and it's duration. To obtain the phonemes asociated with the audio signal was used an ortographic-phonetic transcriptor developed by our group wich is able to obtain the phonemes asociated with a word. To obtain the duration of each of the phonemes a speech segmentation tool was used. This tool, from an audio signal, the signal generated by the TTS system in this case, and from the audio signal's phonetic transcription, gives the phonemes duration. On the other hand, a set of animations for character's lips was generated. To generate these animations the different phonemes, of castilian language in this case, where grouped in base to the place of articulation phonological distinctive feature. Seven different animations where generated according to the next features; bilabial, labiodental, linguodental, alveolar, palatal, velar and glottal. Once we have all the needed elements, a simple method was used to get synchronization, when a word sequence was uttered by the TTS system the associated sequence of movements was generated.
- Text to speech conversion module (TTS): This module produces, artificially, natural language from a word sequence. A commercial system [4] was used.
- Multimodal inputs and outputs: The goal is to overcome the limits that an speech only based system imposes. In a multimodal interaction users are not limited to the use of speech only. Also they have other devices to interact with the system like keyboards, mices, touch screens, etc. In addition, a multimodal system can use different communication channels like speech, text, graphics, etc. In the system presented a screen was used to show text and graphics to the user. Also where used a keyboard and a mouse, with the purpose of being able to browse through the

information shown to the user. The system also has a webcam that is used by the face detection device.

3 Conclusions

This work has been focused on the design of a dialogue system friendly with the user. With this purpose there have been used different input devices in addition to speech. Also have been used different ways of providing information to the user like synthetic speech and graphics.

On the other hand, and thanks to the fact that EDECAN architecture makes able the integration of this kind of systems in any type of device, it's posible to think in the inclusion of a system like the presented in this work in a PDA device. In fact, it has been yet implemented a dialogue system in which the mobile device acts as client of a server that provides the heavy services, like speech recognition, and the mobile device acts as the medium for information displaying.

Acknowledgments

This work has been partially supported by the Counseling of Industry of Basque Government; by the University of the Basque Country under grant GIU07/57; by the Ministry of Science and Innovation under grant TIN2008-06856-C05-01 and by Consolider Ingenio-2010 program MIPRCV (CSD2007-00018).

References

1. Lleida, E., Segarra, E., Torres Barañano, M.I., Macías, J.: EDECÁN: sistEma de Diálogo multidominio con adaptación al contExto aCústico y de AplicacióN. IV Jornadas en Tecnología del Habla, Zaragoza, Spain (November 2006)
2. García, J.E., Ortega, A., Miguel, A., Lleida, E.: Arquitectura distribuida para el desarrollo de sistemas de diálogo hablado, EDECÁN. V Jornadas en Tecnologia del Habla, Bilbao, Spain (November 2008)
3. EDECÁN Project, http://www.edecan.es
4. Loquendo. Vocal technology and sevices, http://www.loquendo.com
5. Open Source Computer Vision, http://sourceforge.net/projects/opencvlibrary
6. Reallusion 3D animation software, http://www.reallusion.com

Integration of Speech and Text Processing Modules into a Real-Time Dialogue System*

Jan Ptáček[1], Pavel Ircing[2], Miroslav Spousta[1], Jan Romportl[2], Zdeněk Loose[2], Silvie Cinková[1], José Relaño Gil[3], and Raúl Santos[3]

[1] Charles University in Prague, Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{ptacek,spousta,cinkova}@ufal.mff.cuni.cz

[2] University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics
Univerzitní 8, 306 14 Plzeň, Czech Republic
{ircing,rompi,zloose}@kky.zcu.cz

[3] Telefónica I+D
Emilio Vargas St., No. 6., 28043 Madrid, Spain
{joserg,e.rsai}@tid.es

Abstract. This paper presents a real-time implementation of an automatic dialogue system called 'Senior Companion', which is not strictly task-oriented, but instead it is designed to 'chat' with elderly users about their family photographs. To a large extent, this task has lost the usual restriction of dialogue systems to a particular (narrow) domain, and thus the speech and natural language processing components had to be designed to cover a broad range of possible user and system utterances.

Keywords: automatic dialog systems, human-computer interaction, speech technologies, natural language processing.

1 Introduction

The COMPANIONS project represents a slightly different approach to automatic dialogue system development. Instead of designing strictly task-oriented system that would robustly cover a limited domain, the goal was to build a system that is more like an artificial 'companion', i.e., able to chat with the user and allowing him/her to develop some 'relationship' with the system. In order to reach this goal, the system is conceived much more broadly, even with the ability to express emotions to a certain degree.

The original plan was to develop a system that would be able to conduct a natural dialogue with elderly users, mostly to keep them company and helping them to stay mentally active. Since this is too broad a scope to be handled, it was decided to narrow the task to reminiscing about family photographs. The system was named 'Senior Companion' and was originally planned to be developed for two languages — English

* This work was funded by the COMPANIONS project (http://www.companions-project.org — EC grant number IST-FP6-034434) and partially also by the projects LC536, ME838, MSM0021620838, GA405/09/0278 and UWB internal project SGS-2010-054.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 552–559, 2010.

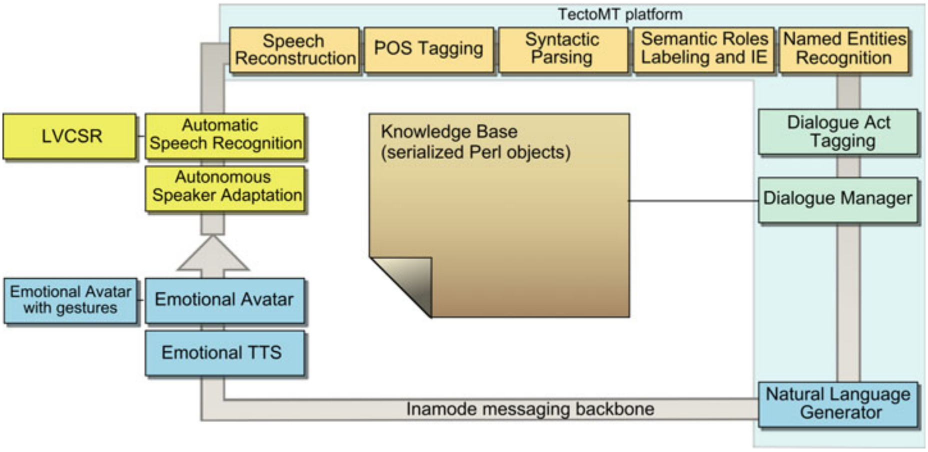

TectoMT platform
Speech Reconstruction
POS Tagging
Syntactic Parsing
Semantic Roles Labeling and IE
Named Entities Recognition
LVCSR
Automatic Speech Recognition
Autonomous Speaker Adaptation
Knowledge Base (serialized Perl objects)
Dialogue Act Tagging
Dialogue Manager
Emotional Avatar with gestures
Emotional Avatar
Emotional TTS
Inamode messaging backbone
Natural Language Generator

Fig. 1. Architecture of the system

and Czech. However, during the course of the project, the domain of the English system has slightly shifted and thus this paper describes the Czech system only.

In this paper, we will focus mostly on the problems related to the implementation and integration of the individual modules of the dialogue system prototype, whilst the scientific background of their development will be mentioned only briefly, with appropriate references to related work.

2 Architecture of the System

The architecture of the system is given in Figure 1. The Czech Companion consists of a number of independent modules (which in principle can reside on different machines) communicating via network messages encoded in XML. It is operating in an endless loop - each cycle of the processing loop starts with a user utterance and ends with the system's reply. There is a central messaging hub which controls the communication among the modules. Each module is geared towards a specific subtask, e.g. automatic speech recognition (ASR), natural language understanding (NLU), Embodied Conversational Agent (ECA) module and others, which are connected via a TCP/IP socket to the messaging hub.

Such an approach allows to employ modules, no matter whether they are Windows-, Linux- or Mac-based and regardless of their programming language. Each module is wrapped into a connector that provides the messaging API. The connectors have been developed and tested for Java, Perl and Python. During the development phase, each partner has been running his/her modules at his/her site while being connected to a central hub over the network to eliminate the installation and remote maintenance efforts. Furthermore, the architecture allows to conveniently relocate any computationally demanding modules to a dedicated hardware. The hub and the connectors are developed by our Spanish project partner (Telefónica I+D) and known under the "Inamode" trademark.

To establish a natural dialogue, it is necessary for the system to be able to respond within a reasonable time (empirically within 3s). In order to achieve such a response time, we have identified modules that work with the same data, and grouped them together into one run-time "supermodule", running on a single machine. For an easy grouping of such modules, we have re-used the TectoMT platform [1], originally designed for the machine translation task. The TectoMT platform architecture provides access to a single in-memory data representation through a common interface, effectively eliminating the overhead of a repeated serialization and XML parsing.

3 Automatic Speech Recognition (ASR)

The speech recognition engine embedded in our application uses state-of-the-art ASR technologies enhanced with some innovative techniques developed in our research labs. Its acoustic model employs Hidden Markov Models (HMMs) trained using large speech corpora (over 220 hours of transcribed speech, more than 700 speakers), which grants a robust acoustic recognition independent of the speaker. This performance is further improved by our original speaker adaptation algorithm, which is completely hidden from the user and does not require the usual explicit offline training session, since it accumulates statistics from all the user's utterances that are recognized with high-enough confidence score. Once the amount of "confidently recognized" data is sufficient, the system estimates the feature MLLR adaptation matrices and uses them to transform the input speech vectors from that point onwards [2].

Another notable feature of our ASR system is the speech decoder itself. It is currently able to handle a lexicon with more than half a million words. The decoder works with standard n-gram language models and allows 2-pass recognition process where the bigram model is employed in the first pass and the resulting lattices are re-scored with higher-order n-grams in the second pass. Note that despite of this 2-pass technique the decoder still operates in real-time [3].

4 Speech Reconstruction

Instead of feeding the ASR output to the Natural Language Understanding modules for further processing directly, we have inserted an intermediate step called Speech Reconstruction. The reason is that NLU modules are typically trained on data coming from annotated text corpora; ours are no exception. Such corpora are usually based on newspaper text and differ significantly from the style, lexicon and register of a spontaneous speech dialogue. Moreover, in such dialogues it is very common to encounter ungrammatical sentences, speaker auto-corrections, repetitions and other irregularities, which are rarely present in written text, resulting in poor results of the subsequent NLU. The speech reconstruction module aims to transform the ASR output into standard grammatical sentences suitable for NLP tools by removing disluencies, changing word order, or even colloquial morphemes. Our system employs machine translation approach; we have trained the Moses statistical machine translation system to "translate" the recognized speech output into fluent written-text-like utterances. We have used 45.000 sentences from the manually edited and corrected ("reconstructed")

PDTSL corpus [4] to train Moses' translation model and a 10-million-word textual corpus for its language model.

5 Natural Language Understanding

The natural language understanding (NLU) pipeline starts with part-of-speech (POS) tagging. Its result is very important in the subsequent steps, such as dependency parsing or named entity recognition, as they rely on correct POS tags. The POS tagging software for the Czech language uses a large morphological dictionary [5] that assigns a set of possible POS tags to every word. Then, a machine learning algorithm is used to select the correct tag. The state-of-the-art tagger of Czech achieves 96% accuracy on PDT 2.0 test set. We have further enhanced the coverage of the dictionary on the Wizard-of-Oz training corpus [6].

Then, we use a robust dependency-oriented syntactic parser; its task is to assign every word its parent in the syntactic dependency tree of the input sentence. The state-of-the-art algorithm [7] is based on the Maximum Spanning Tree algorithm and trained on the PDT 2.0 [8] analytical layer. Its accuracy on the standard PDT 2.0 test set is 85%.

Parsing is followed by a semantic analysis the output of which is a "tectogrammatical" transform of the dependency parse tree, simplified in its structure but enriched by various semantic attributes at its nodes. The semantic parsing involves (a) the assignment of one of the 69 semantic roles, (b) coordination detection, (c) argument structure assignment, (d) partial ellipsis resolution and (e) pronominal anaphora resolution. These attributes are filled using the *fnTBL* toolkit trained on the PDT 2.0 corpus. Post-parsing detection and correction of ungrammatical edges caused by long user utterances is rule-based.

The resulting semantic tree is matched against tree fragments using a tree querying engine PML-TQ [9]. The queries cover topics from the Dialogue Manager (DM; see below). The DM then stores the extracted information in a form of Perl objects, which are eventually saved on disk after each dialogue turn.

The DM also needs to know which named entities, such as personal names, appear in a given user utterance. The Czech SVM-based Named Entities Recognizer [10] we use achieves standard NER F-measure of 70%.

6 Dialogue Manager

The dialogue is driven by a Dialogue Manager (DM) component developed for the Czech Senior Companion. We build upon the idea and implementation of Field et al. [11], modeling the flow of the dialogue by a set of hand-crafted transition networks. We have re-implemented Field's DM and modified it to suit the type of dialogues as observed in our collected data.

The basic abstraction is a collection of interconnected transition networks, each representing a single topic. The nodes in a network represent various states of the dialogue, while the arcs provide possible continuations of the dialogue. Each arc bears a *test* and *action*. In our implementation, both are Perl code statements. The test is used to find out which arcs are suitable continuations given the current dialogue state. Once a continuation arc is selected, its action code is executed. The action performs calls to DM API, updating DM's data structures to reflect the new dialogue state. Finally, the action assembles a "recipe" (a formal structure describing the content of the system response) for the NLG component to generate the system reaction in plain text. Several arcs may be traversed before an action orders the DM to send all the collected "recipes" to the NLG module.

We use an additional layer of Perl objects added between the DM and the NLU datastructures. These objects (persons, events, photos) model the knowledge acquired in the course of the dialogue and provide basic reasoning as well (e.g. it can tell age from a date of birth). Each object's property is able to store multiple values with varying level of confidence, and values restricted to a defined time span. For each topic, we have gathered a number of questions and remarks the system is supposed to say when it has the initiative, as well as replies to anticipated user turns. Here the large amount of collected dialogues has proved to be an irreplaceable resource, despite being used "only" for manual tuning of the DM networks. There is one initialization arc for each network, which is always traversed during each sub-dialogue in the active topic. The initialization arc leads to a central node as in Figure 2. The eccentric beams rooted in the central node take care of the various sub-dialogues. A sub-dialogue is picked at random, taking into consideration past traversals and of course making sure that the test on the entry arc of a sub-dialogue holds. Once the end of the sub-dialogue beam is reached, the DM is redirected back to the initialization arc.

The DM can also handle simultaneous dialogues with multiple users connected over the Inamode interface or over the XMPP/Jabber instant messaging protocol. The user-dependent data are grouped into objects, and the DM keeps track of the user by using an additional level of indirection. Prior to processing every incoming message, a pointer is updated to point to current user's data.

7 Natural Language Generation

Natural Language Generation (NLG) is implemented as a module of its own right, instead of a simple template approach typically used in previous dialogue systems. There are several reasons for that: Czech features a very rich inflection and morphosyntactic agreement (also related is its pro-drop feature), and its "free" word order used to convey emphasis and the topic-focus articulation. The NLG accepts a (grammatically grossly underspecified) tree-shaped template, which contains only the content to be formulated in the output natural language. These tree templates (called "recipes" in the previous section) are collected from the edges in the DM's transition network. They use linearized syntax that draws from the Graphviz dot notation [12]. The notation describes the tree by a set of statements specifying node properties and/or dependency relations between them. The integration with the DM is implemented using references to DM's Perl objects. Each reference is expanded to a set of additional statements, providing the subsequent NLG proper with the lexically and morphologically important features (e.g. gender) of those objects.

The rule-based sentence generation process consists of a sequence of linguistically motivated steps: "formeme" selection, morphological agreement, addition of functional

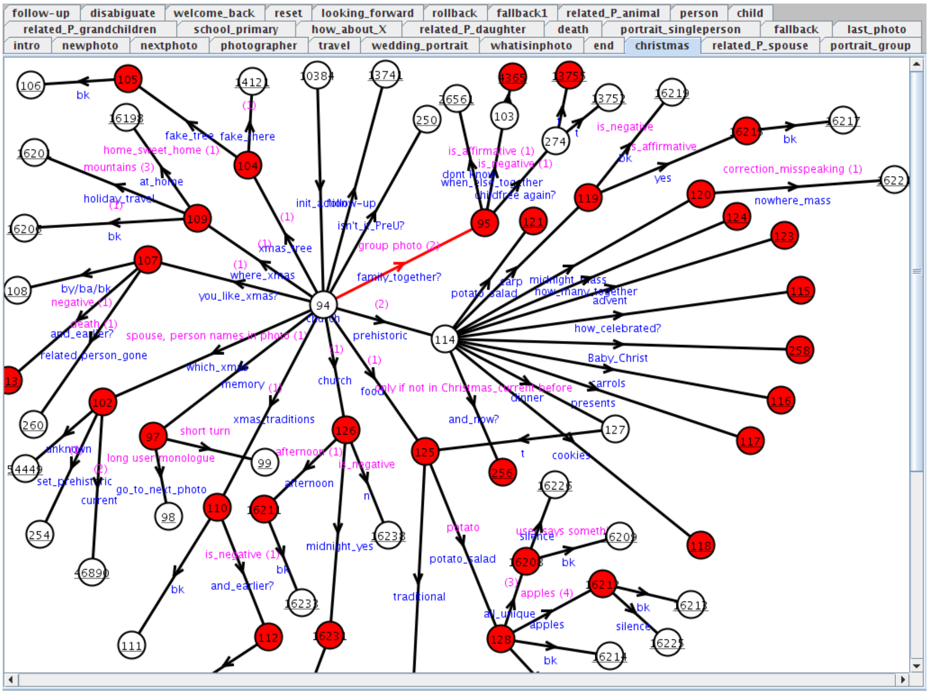


Fig. 2. Star patterned transition network for the topic of *Christmas* (red nodes mark a change in initiative)

words, inflection, word ordering, punctuation, and capitalization. The formeme selection phase is where the sentence takes on its surface-syntactic shape: the expanded input tree is traversed in a depth-first fashion, and a suitable morphosyntactic form is selected from the repertoire of forms available in Czech (i.e. prepositional phrase with the correct case, direct case, infinitive form, etc.). Once the formemes are established, the tree is processed by a cascade of operations that create the form selected and encoded by the formeme, so that the rules of Czech grammar are obeyed. This process includes, i.a., addition of functional words, proper morphological adjustments based on person, number and gender and realization of correct verb tenses. Finally, the word order is generated, as well as the correct punctuation and capitalization.

8 Embodied Conversational Agent

The system talks to the user through a graphically embodied conversational agent (ECA) which consists of a text-to-speech (TTS) system and a visual representation of a human character (the "avatar").

The TTS system is based on the Czech state-of-the-art speech synthesis system ARTIC [13] which employs the unit selection synthesis technique. Since one of the main goals of the COMPANIONS project is to enhance the naturalness of human-computer interaction as much as possible, the usage of a high quality emotional TTS voice is essential. The ARTIC voice used for this purposes is based on an expressive speech corpus specially created for the COMPANIONS project using the Wizard-of-Oz corpus and a communicative function annotation scheme. Since it is difficult to classify human speech according to categorical or dimensional emotion models, we have settled for the assumption that a relevant affective state of ECA goes implicitly together with a *communicative function* of a speech act (or utterance), which is more controllable than the affective state itself. It means that we do not need to think of modeling an emotion such as 'guilt' per se, we expect it to be implicitly present in an utterance like 'I am so sorry about that' with a communicative function '(affective) apology'.

Apart from the TTS system, the ECA features a visual avatar. The final version of the Czech Senior Companion makes use of the Telefónica I+D avatar. It is a graphical visualization of a female head and torso with the capability of articulation, facial expressions and body gestures, which are triggered by special commands with both categorical and continuous parameters. A window with the avatar can be embedded e.g. in a web browser, which means that this ECA is able to run in various environments and modes of usage. The avatar as well as the TTS system is connected to the central messaging hub, as described in the Section 2. Both modules accept incoming messages and send outgoing messages to other modules. The activity of the avatar and TTS modules is coordinated by a proxy module. The proxy module accepts NLG output messages - every such a message contains text of the whole dialogue turn of ECA and the communicative functions of all the sentences in this text, as assigned by the Dialogue Manager. The proxy module parallelizes the work-flow by allowing the TTS system and the avatar to work simultaneously: TTS synthesizes sentences one by one and sends them to the input buffer of the avatar while the avatar is playing them stepwise. This measure was employed to minimize the latency of the system's response. The proxy module also communicates with a gesture module in order to transform the communicative functions assigned by DM into appropriate commands triggering gestures and facial expressions of the avatar. This process involves certain randomness causing the visual behavior of ECA be more natural. Synchronization of the avatar's articulation with synthesized speech is ensured by the TTS module, which generates an accompanying string of phones and their time-stamps.

9 Conclusions

The presented implementation shows that even the most advanced speech and natural language processing technologies can be successfully transferred from a laboratory form to a software prototype that is working in real-time. However, it is clear that the various modules can still be enhanced to avoid errors that still occur, such as those caused by the Named Entity Recognizer or the co-reference module. Also, the speech reconstruction module is far from perfect it can handle simple disfluencies but fails on more complicated edits. Finally, the DM should be trained as well, using e.g. the technique of Partially Observed Markov Decision Processes (once we are able to collect more dialogue data by using our Wizard-of-Oz technique). In fact, the resulting dialogue corpus is and will be one the most valuable results of the project, at least for the Czech language; it can and certainly will serve to advance research on general enhancement and domain-adaption all of the speech and natural language analysis and generation components (not just the DM).

Dialogue systems are difficult to evaluate, especially if not task-oriented. A preliminary evaluation done by both the COMPANIONS consortium partners and the project reviewers rated our system's overall performance to be very good. However, a proper evaluation suited for free-flowing conversation still remains to be done. Such evaluation is actually one of the primary goals of the final year of the COMPANIONS project.

References

1. Žabokrtský, Z., Ptáček, J., Pajas, P.: TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In: StatMT 2008: The Third Workshop on Statistical Machine Translation, Morristown, NJ, USA, pp. 167–170. ACL (2008)
2. Zajíc, Z., Machlica, L., Müller, L.: Refinement approach for adaptation based on combination of MAP and fMLLR. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 274–281. Springer, Heidelberg (2009)
3. Pražák, A., Müller, L., Psutka, J.V., Psutka, J.: Live TV subtitling - fast 2-pass LVCSR system for online subtitling. In: SIGMAP 2007, Barcelona, pp. 139–142 (2007)
4. Cinková, S.: Semantic representation of non-sentential utterances in dialog. In: SRSL 2009, Morristown, NJ, USA, pp. 26–33. ACL (2009)
5. Hajič, J.: Disambiguation of Rich Inflection (Computational Morphology of Czech). Nakladatelství Karolinum, Prague (2004)
6. Grůber, M., Legát, M., Ircing, P., Romportl, J., Psutka, J.: Czech senior COMPANION: Wizard of Oz data collection and expressive speech corpus recording. In: LTC 2009, Poznan, Polan, pp. 266–269 (2009)
7. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: HLT 2005, Morristown, NJ, USA, pp. 523–530. ACL (2005)
8. Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M.: Prague Dependency Treebank v2.0, CDROM, LDC Cat. No. LDC2006T01. Linguistic Data Consortium, Philadelphia, PA (2006)
9. Pajas, P., Štěpánek, J.: PML Tree Query 0.5 alpha (2008), `http://ufal.mff.cuni.cz/~pajas/pmltq/`
10. Kravalová, J., Žabokrtský, Z.: Czech named entity corpus and SVM-based recognizer. In: NEWS 2009: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, Morristown, NJ, USA, pp. 194–201. ACL (2009)
11. Field, D., Catizone, R., Cheng, W., Dingli, A., Worgan, S., Ye, L., Wilks, Y.: The senior companion: a semantic web dialogue system. In: AAMAS 2009, Richland, SC, pp. 1383–1384 (2009)
12. Ellson, J., Gansner, E.R., Koutsofios, E., North, S.C., Woodhull, G.: Graphviz - open source graph drawing tools. In: Graph Drawing, pp. 483–484 (2001)
13. Tihelka, D., Matoušek, J.: Unit selection and its relation to symbolic prosody: a new approach. In: INTERSPEECH 2006, Pittsburgh, PA, pp. 2042–2045 (2006)

Client and Speech Detection System for Intelligent Infokiosk

Andrey Ronzhin[1], Alexey Karpov[1], Irina Kipyatkova[1], and Miloš Železný[2]

[1] St. Petersburg Institute for Informatics and Automation of RAS
SPIIRAS, 39, 14th line, St. Petersburg, Russia
[2] University of West Bohemia
Univerzitní 8, Pilsen, Czech Republic
{ronzhin,karpov,kipytkova}@iias.spb.su, zelezny@kky.zcu.cz
http://www.spiiras.nw.ru/speech, http://www.kky.zcu.cz

Abstract. Timely attraction of a client and detection of his/her speech message in real noisy conditions are main difficulties at deployment of speech and multimodal interfaces in information kiosks. Combination of sound source localization, voice activity and face detection technologies allowed to determine client mouth coordinates and extract boundaries of speech signal appeared in the kiosk dialogue area. Talking head model based on audio-visual speech synthesis immediately greets the client, when her face is captured in the video-monitoring area, in order to attract him/her to the information service before leaving the interaction area. Client's face tracking is also used for turning the talking head in direction to the client that significantly improves the naturalness of interaction. The developed infokiosk set in the institute hall provides information about structure and staff of laboratories. Statistics of human-kiosk interaction is accumulated within last six months in 2009.

Keywords: Multimodal interfaces, information kiosk, sound source localization, face tracking, talking head, voice activity detection.

1 Introduction

In last decade, various smart information kiosks (infokiosks) with speech and multimodal user interfaces have being actively developed and studied. The following systems have to be mentioned among good samples of intelligent kiosks: Touch'n'Speak system developed by the Tampere University (Finland); Memphis Intelligent Kiosk Initiative (MIKI) [1] from the Memphis University (USA); French system Multimodal-Multimedia Automated Service Kiosk (MASK); Multimodal Access to City Help Kiosk (MATCHKiosk) [2] manufactured by AT&T company. An information input can be provided by a touchscreen/keyboard and by voice and even by manual or body gestures. A multimodal analysis of human being behavior is the most perspective way that gives complete information about activity of the person, therefore in many recent studies the audio-visual analysis is integrated in one system of smart space [3,4].

A prototype of an infokiosk called MIDAS (Multimodal Interactive-Dialogue Automaton for Self-service) has been developed and installed in SPIIRAS. The infokiosk

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 560–567, 2010.

is able to detect a human being inside the working zone with the Haar-based object detector [5], track his/her movements and demonstrate awareness using a visual feedback by a 3D avatar, which tracks clients by rotation of his talking head [6]. The infokiosk realizes a mixed initiative dialogue strategy with a client, starting from the system's initiative and giving initiative for the query to a new user after a verbal welcome.

Chapter 2 describes general architecture of the proposed multimodal automaton, configuration of microphone array, video cameras, software modules and base scenarios of the kiosk work. Experimental results of sound source localization are presented in Chapter 3. An analysis of multimodal interaction with MIDAS kiosk is reported in Chapter 4.

2 Technological Framework of the Multimodal Infokiosk

While developing the multimodal kiosk there is an important problem of interconnecting several algorithms, recognition and signal processing technologies that work on data streams incoming from a high number of sensors from different modalities. Any information system can be divided into several basic parts: central control unit, input sensors, and output part. Input sensors comprise a visual part, an acoustic part, and a touch screen part. An output is presented to the user using graphical screen. Central control unit is a multimedia PC equipped with special hardware for recording and processing of the input data. Central control unit also takes care of the connection to external data sources, such as various types of databases. To allow efficient communication of sensor input and output data it is useful to have a middleware layer that provides infrastructure services.

The general architecture of the information kiosk and description of hardware components and software technologies, which work simultaneously and synchronously, are considered in [7]. The most important of them are: (1) video processing using a technology of computer vision in order to detect position of human's body, face and some facial organs; (2) speaker-independent system of automatic recognition of continuous Russian speech that uses a microphone array to eliminate acoustical noises and to localize a source of useful voice signals at distant speech recording; (3) a module for Russian speech synthesis to be applied for realization of a virtual talking character – avatar [6]; (4) an interactive graphical user interface with a touchscreen; (5) a dialogue manager that has access to an applied domain database and synchronizes sub-modules for input modalities fusion and output modalities fission.

Fig. 1 shows a scheme of the location of the information input/output devices of the interface part of the automaton. An array of 4 microphones Oktava MK-012 was designed for the kiosk. Three microphones are positioned on one horizontal line under the touchscreen and one microphone is placed above the touch screen. Such configuration allows calculating 3D coordinates of sound sources. It also helps record clean user's speech because several microphones are located near user's mouth and surrounding noises are suppressed by the human's body. One portable web-camera Logitech QuickCam Pro is placed above and another below the touchscreen (non stereo-pair of the cameras). At that the upper web-camera is tilt down and set in 5 cm into the kiosk's construction (the same as one upper microphone), the lower camera was

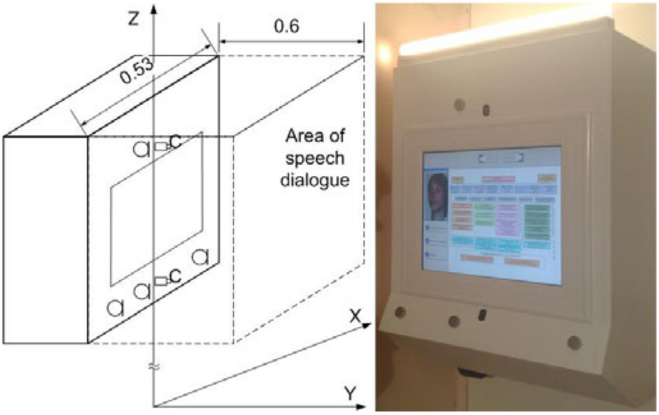


Fig. 1. Scheme of location of the input/output devices and view of the kiosk

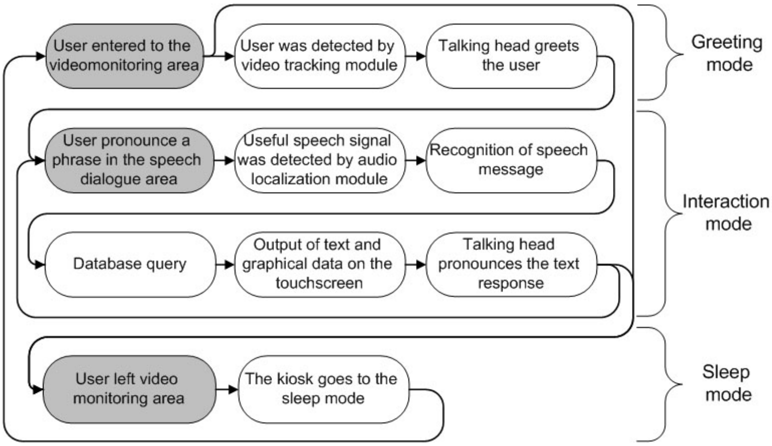

User entered to the videomonitoring area
User was detected by video tracking module
Talking head greets the user
Greeting mode
User pronounce a phrase in the speech dialogue area
Useful speech signal was detected by audio localization module
Recognition of speech message
Interaction mode
Database query
Output of text and graphical data on the touchscreen
Talking head pronounces the text response
User left video monitoring area
The kiosk goes to the sleep mode
Sleep mode

Fig. 2. General scenario of the kiosk's work

rotated up to 30 degrees. This configuration allows keeping user's face with any height in the field of vision. The working zone, where head and mouth of a potential user are searched, is 60 cm in front of the kiosk i.e. at the average arm's length. The implementation of two types of sensors for user detection allows one to improve the quality and decrease influence of audio- and video noises. The microphone array is used for sound source localization and two web-cameras are used for detection and tracking of user faces. For timely attraction of clients, the area of video-monitoring was extended up to 2.5 m from the kiosk. Talking head model immediately greets the client, when her face is captured in the video-monitoring area, in order to attract him/her to the information service before leaving the interaction area.

The base scenarios of kiosk work with three modes (greetings, interaction, sleeping) in correspondence with the user actions is presented in Fig. 2. The common cases of user-kiosk interaction are: (1) user gone past the kiosk too fast in order to be captured by the module of video-monitoring; (2) user gone into the video-monitoring area, the talking head produced greeting, but he/she gone away; (3) user gave a speech command inside the speech dialogue area, sound source was localized, useful speech signal was recognized, corresponding database query was accomplished and found data were outputted on the touch-screen and generated by the talking head, after that the user continued the dialogue or gone away. During the interaction, the module of video tracking seeks user faces on each captured frame and rotates the talking head to the direction of a greatest face. The talking head follows the user and shows own awareness about user presence that improves naturalness of the interaction and attracts new users to the information service.

3 Multichannel Audio Processing for Spatial User Localization

While developing and embedding a speech interface to any application there are lots of difficulties connected with its usability and reliability. In order to solve these problems the combination of a speaker-independent speech recognition system with a microphone array and a noise elimination technique can effectively be applied [8]. Rather good results for distant speech recognition (with distance 0.5–1.0 m from a microphone) are obtained by a directed linear microphone array with a noise suppression system. Usage of an array consisting of several microphones demonstrates advantages of binaural hearing. Even two microphones in the array allow perceiving more quiet sounds that is very important for evaluation of the surrounding sound space.

Additionally to the function of acoustical signals capturing the microphone array can be employed for spatial localization of sound sources. Taking into account that position of a human being in front of the kiosk is predefined, then the microphone array has to perceive just audio signals incoming from the frontal working zone and eliminate all sounds both from side and back of the automaton. Efficiency of the microphone array strongly depends on its geometry and beamforming algorithms, which are used for combining signals incoming from different microphones as well as relies on the methods for filtering and suppression of acoustical noises.

When processing a speech signal using two or more microphones, acoustic spatial estimation ability of binaural hearing is applied. Besides, immediate function of signal receiving, the sensor array is used for source spatialization and filtration of desired signals. In the latter case, beamforming techniques are used, those make the system receive and process only signals radiated from a narrow area and minimize (or cut off) signals coming from other directions. For tasks when only source localization without spatial filtration is needed, Time Delay Estimator technique is used. General Cross-Correlation (GCC) method [9] or signal phase processing [10] and [11] are used in the majority of applications for estimation of time-delay of sound wave arrival. All these methods imply localization of the speaker with the help of a set of delay estimates, calculated on different microphones pairs. The coordinates of the sound source are calculated with the aid of the triangulation method, and estimates of signal duration and power spectrum indicate whether speech is present in the recorded audio signal.

The developed method for spectral-spatial analysis of speech activity is based on GCC-PHAT (phase transform) and using the microphone array with the “reversed T” form. The determination of correlation maximum of mutual spectrum of signal pairs, recorded by different microphones, allows to evaluate phase difference between the signals. Then calculation of coordinates of a sound source is made by the triangulation method. The energy level of mutual signal spectrum and estimation of acceptable position of a speaker is used for finding the boundaries of speech in a multi-channel acoustical stream, recorded in noisy environments.

In the framework of MIDAS, the X-axis runs along lower boundary of front panel of the kiosk. The Y-axis goes from the center of kiosk and towards a user; the Z-axis goes vertically via center of the front panel of the kiosk. Thus, the origin of coordinates is located in the middle on lower boundary of front panel of the kiosk, as it is showed Fig. 1. The given plots proof that a working area can be bounded across the width of the body of kiosk (53 cm) and 60 cm on Y-axis; this would be sufficient for a comfortable use of the touch-screen. As for the Z-axis, the working zone can be at the distance from 120 cm till 180 cm from the floor.

In the course of the first experiment, a set of 2,500 phrases was played through a loudspeaker mounted 30 cm in front of the kiosk for estimation of 3D localization accuracy of the algorithm. Fig. 3 illustrates calculated values of loudspeaker's coordinates, which were estimated during signal recording. It can be seen that majority of speech segments were localized with the same coordinate values (−0.06; 0.32; 1.55). Some scatter of parameters is present around the Y-axis. Nevertheless, applying methods of smoothing can enhance procedure of coordinate estimation.

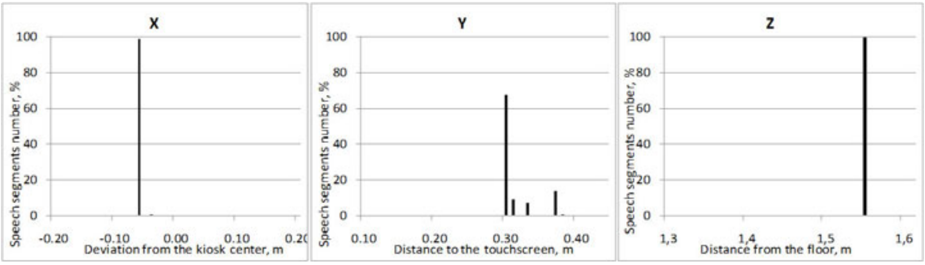


Fig. 3. Values of loudspeaker's coordinates estimated automatically

In the course of the next experiment, phrases were spoken by prepared users. Given that parameters of the system were kept the same, growth of scatter is owing to natural short movements of users in front of the kiosk. Fig. 4 illustrates values of user's mouth coordinates, which were estimated in the course of recording. 5 users with various heights participated in the experiment, each of them told 500 phrases. Total duration of speech segments without pauses between phrases was 51 minutes.

The complete version of the infokiosk was set in the hall of the institute and the experiments with audio localization were continued. Fig. 5 presents the results of localization of phrases pronounced by more than one hundred different unprepared users. The mean values of user mouth coordinates left similar previous experiment, but the variety was increased. Nevertheless, the selected size of speech dialogue area was satisfactory for all the users.

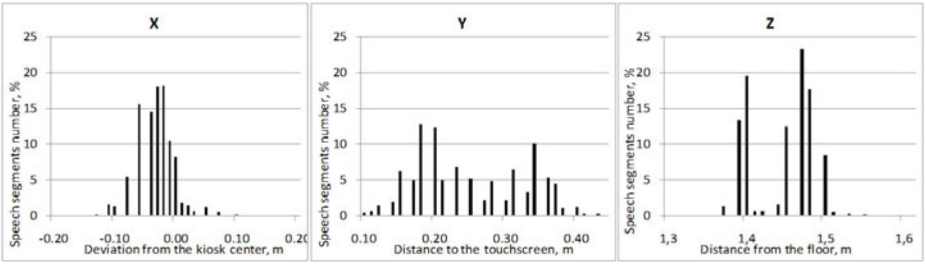


Fig. 4. Values of users' mouth coordinates estimated automatically

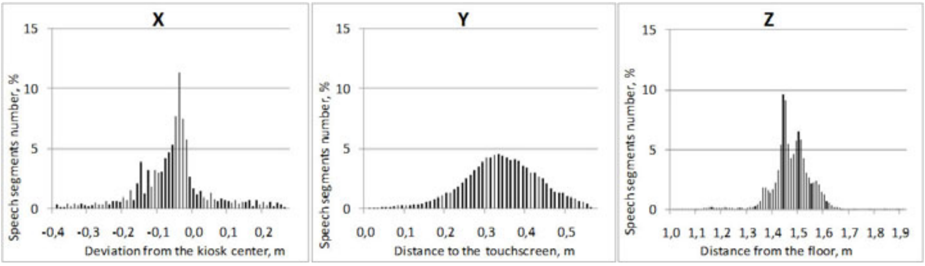


Fig. 5. Users' mouth coordinates estimated automatically during half year of kiosk use

No children took part in the experiments, but Z-axis can be lowered for children's comfort. It was found out during the experiments that usually the user bends forward to the kiosk and raises her voice intuitively to increase the loudness of the signal.

4 An Analysis of Multimodal Interaction

The infokiosk provides information about the structure and staff of laboratories, location and phones of departments and employees, as well as contact information useful both for visitors and employees of SPIIRAS. To access the interactive diagram of the institute structure, users may apply both touchscreen and/or voice requests. Automatic Russian speech recognition uses the typical HMM-based approach and a finite-state grammar with a small-sized vocabulary of over 400 words. Statistics of HCI is based on analysis of system's logs, voice dialogues and a gallery of photos of users within six last months in 2009 in the automated mode with WOZ (in the automatic mode in off hours). All users could easily observe video-cameras and microphones of the kiosk but they were not informed about on-line audio-visual recordings. Table 1 summarizes main statistics of multimodal HCI.

It was discovered most frequent inquiries were related to information about "Director" and "Library" giving almost 20% of all the voice questions. Often users asked contacts of deputy directors and heads of departments. However names of departments were rarely pronounced (less than 5%). It is quite difficult to differentiate user's sessions

Table 1. Statistics of multimodal interaction

Quantitative Indicator	Value
Avg. sessions per day	4.20
Unique users (based on face analysis)	145
Avg. speech inputs per session	2.77
Avg. informative voice queries / session	1.80
Sentence recognition rate by ASR	55%
Dialogue completion rate with WOZ	96%
Requests by the touchscreen	1,018

that were caused by real needs and sessions with the aim to play with new technologies. For example, in some questions users asked names who have not any relations to the information service, or other out-of-vocabulary (OOV) words and uninformative messages. Total number of OOV words pronounced was approx 15%; they were misrecognized by ASR, but the human-corrector modified kiosk's answers in most of such cases.

5 Conclusion

Combination of audio source localization, voice activity detection and face tracking technologies was realized in the developed multimodal infokiosk equipped by the standard means for information input/output (touch-screen and loudspeaker) and the devices for contactless HCI (microphone array and web cameras). The kiosk able to determine the client's mouth coordinates and to detect boundaries of speech signal appeared in the kiosk speech dialogue area. The size of the speech dialogue area was selected and tested during the experiments with prepared users and at real exploitation of the kiosk. The talking head model immediately greets the client whose face is captured in the video monitoring area to attract him/her to the information service. The talking head is used for audio-visual speech synthesis. It also follows the user and shows own awareness about user presence that improves the naturalness of the interaction.

Acknowledgments. This work is supported by the Russian Federal Targeted Program (contracts No. 2360 and 2579), Russian Foundation for Basic Research (projects No. 08-08-00128-a and 08-07-90002-CT-a), Czech Republic Ministry of Education (project No. ME08106), as well as by the Human Capital Foundation.

References

1. McCauley, L., D'Mello, S.: MIKI: a Speech Enabled Intelligent Kiosk. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 132–144. Springer, Heidelberg (2006)
2. Johnston, M., Bangalore, S.: MATCHkiosk: A Multimodal Interactive City Guide. In: Association of Computational Linguistics ACL 2004, Barcelona, pp. 223–226 (2004)

3. Gatica-Perez, D., Lathoud, G., Odobez, J., McCowan, I.: Multimodal Multispeaker Probabilistic Tracking in Meetings. In: 7th International Conference on Multimodal Interfaces ICMI 2005, pp. 183–190. ACM, Trento (2005)
4. Zhang, C., et al.: Boosting-Based Multimodal Speaker Detection for Distributed Meeting Videos. MultMed. 10(8), 1541–1552 (2008)
5. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. In: 9th IEEE International Conference on Image Processing ICIP 2002, Rochester, New York, pp. 900–903 (2002)
6. Karpov, A., Tsirulnik, L., Krnoul, Z., Ronzhin, A., Lobanov, B., Zelezny, M.: Audio-Visual Speech Asynchrony Modeling in a Talking Head. In: 10th International Conference Interspeech-2009, pp. 2911–2914. ISCA, Brighton (2009)
7. Karpov, A., Ronzhin, A.: Information Enquiry Kiosk with Multimodal User Interface. Pattern Recognition and Image Analysis 19(3), 546–558 (2009)
8. Brandstein, M., Ward, D.: Microphone Arrays. Springer, Heidelberg (2000)
9. Knapp, C., Carter, G.: The Generalized Correlation Method for Estimation of Time Delay. IEEE Transactions on Acoustics, Speech and Signal Processing 24(4), 320–327 (2003)
10. Lathoud, G., McCowan, I.: A Sector-Based Approach for Localization of Multiple Speakers with Microphone Arrays. In: ISCA Workshop on Statistical and Perceptual Audio Processing SAPA 2004. ISCA, Jeiu (2004)
11. Omologo, M., Svaizer, P.: Acoustic Event Localization Using a Crosspower-Spectrum Phase Based Technique. In: International Conference on Acoustics, Speech and Signal Processing ICASSP 1994, pp. 273–276. IEEE, Australia (1994)

Prototype of Czech Spoken Dialog System with Mixed Initiative for Railway Information Service

Jan Švec and Luboš Šmídl

Center of Applied Cybernetics, Department of Cybernetics,
Faculty of Applied Sciences, University of West Bohemia, Pilsen, 306 14, Czech Republic
{honzas,smidl}@kky.zcu.cz

Abstract. This paper describes a prototype of a Czech dialog system with a mixed dialog initiative and a natural language understanding module. The described dialog system is designed for providing railway information such as arrivals, departures, prices and train types. The dialog can be driven by both an user of the system and a dialog manager to accomplish the dialog goal. In addition the user can use an almost arbitrary Czech utterance consistent with the dialog domain to interact with the system. The system accesses the train database on-line via the Internet. The version described in this paper works as a desktop computer application and communicates with the user using the headset. The paper describes the modules of the dialog system including automatic speech recognition, natural language understanding, dialog manager, speech generation and speech synthesis.

Keywords: spoken dialog system; language understanding; mixed initiative

1 Introduction

The spoken dialog systems have many real-world applications including the remote database access or the technical support. The current applications are mostly based on the VoiceXML language. In this language the semantics of the user's utterance is generated by the speech recognition grammars. Therefore the form of the utterance is restricted to the language accepted by the recognition grammar. The dialog itself often has a finite number of states and the dialog flow is driven by the system initiative. In the VoiceXML-based dialog systems the user has a very limited freedom of choice – for example he is not able to choose which information he will present to the system first and which further.

Currently the spoken dialog systems of a new generation are developed and the number of applications of such systems is increasing. These systems allows the dialog with a mixed initiative - the user can provide his request to the system (user initiative) and then the system asks for missing information (system initiative). Then the system queries the database and presents the result to the user. Additionally the user can change the query or request the next or the previous result from the database (the user initiative again). The next-generation dialog systems usually employ the natural language understanding module to yield the semantic representation of the user's utterance. Then the form of the user's utterance is not restricted and the user can use

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 568–575, 2010.

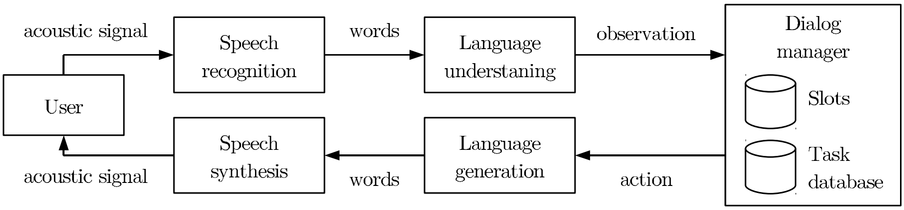


Fig. 1. Scheme of spoken dialog system

an almost arbitrary utterance consistent with the dialog domain. For English there are some spoken dialog systems of the next generation. For example the laboratory dialog system for tourist information domain developed at the Cambridge University [1] or the commercial technical support dialog system developed by SpeechCycle [2].

The Railway information system spoken dialog system (RIS-SDS) described in this paper is a prototype of the Czech next-generation spoken dialog system with mixed initiative, a natural language understanding, and a large task database. The prototype provides railway information for a given train connection such as time of arrival, departure, type of train or ticket prices. The system is able to recognize and understand a natural speech constrained only by the domain of the dialog. The task database is on-line accessed via Internet and the query can be arbitrary constrained by the user. Additionally the user can request information about the next or the previous result from the database.

The scheme of the presented RIS-SDS is depicted on Figure 1. The user of the system uses speech to interact with the dialog system. The speech is recognized using an automatic speech recognition module (ASR) [3]. The output sequence of words is processed in a natural language understanding module which produces a semantic representation of an utterance (the observation). Then a dialog manager uses this observation together with its state information to generate an action which is a formal description of an answer. The action is then converted into a corresponding textual representation in a language generation module and the text is synthesized in a text-to-speech module.

The paper is organized as follows. The ASR module and its parts—the acoustic and language models—are described in Section 2. The language understanding module is described in Section 3. Section 4 covers a dialog management module. Section 5 depicts a language generation and a speech synthesis module. Finally Section 6 concludes the paper.

2 Automatic Speech Recognition

We used real-time large vocabulary continuous speech recognizer to achieve very high degree of interactivity. The user is able to barge-in into the prompt of the dialog system to stop the speech synthesis and change, accept, reject or confirm the slot values immediately. It improves the user experience because even if the speech recognizer introduces an error the user is able to quickly recover this error and continue in the normal dialog flow. The two following sections describe the acoustic and the language model of the ASR module.

2.1 Acoustic Model

The acoustic models were trained on microphone-based high-quality speech corpus. This corpus contained more than 800 speakers and yielded totally about 220 hours of speech. The PLP parameterization was used with 27 filters and 12 PLP cepstral coefficients with both delta and delta-delta sub-features. Therefore one feature vector contains 36 coefficients and it was computed each 10 milliseconds. The individual basic speech unit was represented by a three-state HMM with a continuous output probability density function assigned to each state. Totally 4,922 clustered states each with 16 mixtures of multivariate Gaussians were used (see [4] for details).

2.2 Language Model

The automatic speech recognition module of the dialog system uses the class-based bigram language model. The parameters of the model were estimated from the Czech Human-Human Train Timetable (HHTT) corpus [5]. The corpus consists of 6.353 dialogs recorded in the information center of Czech train operator. The dialogs were manually transcribed and annotated with a semantic annotation. Both the user and the operator of the information center were transcribed. To train the language model we used just the utterances of the user (26k utterances, 194k tokens). Unfortunately the audio recording of the dialogs were recorded from an analog telephone line (single channel for both the user and the operator) and were unsuitable for evaluating the ASR performance.

First we marked the classes in the training corpus. We used the following classes manually created from the training corpus: *gender* (sir, madam), *time_of_day* (morning, afternoon), *weekday* (Monday, ...Sunday), *month* (January, ...December), *number* (numbers 1 to 99, other numbers do not occur the vocabulary), and *special_day* (tomorrow, weekend). These classes were extended to contain the words not found in the corpus (for example, the class *month* was extended with the months that do not occur in the vocabulary). Because Czech is very inflective language we also added all flection forms of the given class member into the same class. We did not use separated class for the flection forms to make the language model more robust.

In addition to these six manually created classes we used the following automatically created classes based on the database of Czech train operator: *station* (contains 2,806 station names in Czech Republic) and *train_type* (12 class members representing train types - express, intercity, direct train, local train, ...including flection and colloquial forms). To simplify the understanding of the utterance, we generated the following classes: *the_station* (nominative of every member in the *station* class), *fromto_station* (genitive), *toward_station* (accusative), and *in_station* (locative). To obtain the flection forms of the station names we used the prepared database of flection forms for all Czech towns.

In spontaneous language, the short forms of the station names occur very often. For example the station name *České Budějovice* is often shortened to *Budějovice*. Czech speakers also tend to use colloquial variants of station names – the genitive of the station name *České Budějovice* is *Českých Budějovic* with its colloquial variant *Českejch Budějovic*. Therefore we developed a set of rules that generate the shortened

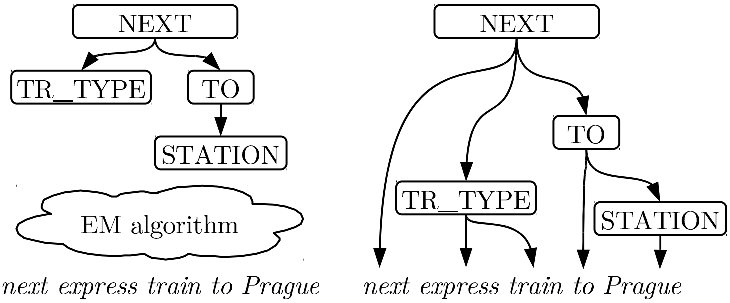


Fig. 2. The abstract semantic annotation (left) and aligned semantic annotation (right)

and colloquial variants for all members of a given class. These variants were added into the pronunciation lexicon as an additional pronunciation of the formal form of station name. This set of rules was applied to the following classes: *the_station* (4,352 pronunciation variants were generated),*fromto_station* (6,143 variants) *toward_station* (5,020 variants) and *in_station* (6,272 variants).

Finally the training corpus was preprocessed and the tokens corresponding to a member of some class were replaced with the class identifier. Then the parameters of a bigram language model were estimated using the Kneser-Ney discounting. All but the station-related classes have uniform probability distribution over class members. The members of all classes corresponding to the station names have assigned empirical probability distribution which depends on the number of citizens of the town in which to the given railway station is located. The total size of the vocabulary is 15,393 words and 4,169 words are not members of any class.

3 Natural Language Understanding

The output of an ASR module is an arbitrary utterance. Therefore we used a hidden vector state (HVS) semantic parser which processes an input utterance and produces a parse tree representing the meaning of the utterance. We used the HVS parser because it is able to estimate the parameters of a model from corpus with abstract semantic annotations. The abstract semantic annotation is more robust than the aligned annotation because the annotators are not forced to align the annotation with the underlaying utterance (for the difference between the abstract and the aligned semantics see Fig. 2). We implemented an extension which allows parsing of sentences with left-right branching structure (LRB-HVS parser) which are very common in Czech [6]. It was shown that the use of this extension can significantly improve the parsing performance of Czech sentences.

For the real-time usage we implemented the LRB-HVS parser as a finite state automaton using an OpenFST toolkit [7]. The states of the automaton represent the stack states of an HVS parser. The starting state of the automaton is equal to an empty HVS stack. It is also a final state of the automaton because the stack of the parser should be empty at the end of the utterance. The weights of the automaton are defined over the tropical semiring and the semantic parsing is equivalent to finding the sequence of states along the shortest path given the input sequence of words.

To make the semantic parser more robust we employed the classes described in Section 2.2. The classes are used to preprocess the semantic corpus before training the HVS parser - the words in the semantic corpus which are members of a given class were replaced with the corresponding class identifier. During the semantic parsing of a given word the HVS parser uses the information of class membership produced by the ASR module.

The result of semantic decoding is the aligned semantic tree (see Figure 2 right). The aligned semantic tree is then converted to the *attribute*:*value* pairs which are more suitable for dialog management. The concept in the root of semantic tree is assigned to an *intention* attribute which represents the action of the user. The other attributes are created from the leaves of the tree and correspond to the slot values. For the semantic tree depicted on Figure 2 the corresponding list of *attribute*:*value* pairs is: [*intention*: next, *train_type*: express, *to_station*: Prague].

The LRB-HVS semantic parser was trained using the Czech HHTT semantic corpus [5]. The semantic layer of the corpus consists of abstract semantic trees assigned to every utterance of both the user and the operator. Totally 1,109 dialogs were annotated by the semantic layer. Exactly one semantic concept is assigned to every node of the semantic tree. The whole corpus uses 35 domain-specific semantic concepts such as DEPARTURE, TIME, STATION, FROM, TRAIN_TYPE etc. The parser was trained on the training set of the corpus (798 dialogs). The performance was measured using the *concept accuracy* [6] evaluated on the test set (223 dialogs). The parser achieved 74.5% of concept accuracy.

4 Dialog Management

After the initial experiments we decided to implement a dialog manager from scratch. The main reason was to design the system that would be able to mix the handcrafted knowledge-based dialog strategy with the statistical methods suitable for dialog management such as partially observable Markov decision processes.

4.1 Dialog State

The dialog state is decomposed into two parts: the set of *slots* and the *task database*. The slots store the information present by an user during the dialog. The slots are of two types: *required* and *optional*. The required slot has to be specified in order to query the task database. The values of optional slots only narrow down the number of results. Each slot has assigned the status: *needed* (the slot is required and has no value), *empty* (the slot is optional and has no value), *filled* (the slot has assigned exactly one value), *confirmed* (the slot value was confirmed by the user) and *overspecified* (the slot has assigned more than one value, its value is inconsistent). The task database represents the real-world data and can be queried for the given object according to the values of the slots. The task database is also able to select a next or previous object matching the query.

4.2 Dialog Strategy

The dialog manager uses two sets during the generation of an action. The first set is called the *pool* and contains semantic representation of all possible actions that can be taken in the given dialog state. The second set is called *handlers* and contains procedural rules that will be executed at the beginning of the next dialog turn after receiving the observation. The basic dialog strategy can be decomposed into three steps executed every dialog turn:

1. *Handle the input* – execute all handler rules added to the set of *handlers* at the end of previous turn and empty the *handlers* set. Then modify the current dialog state according to the value of an *intention* attribute in the observation.
2. *Generate possible actions* – add pairs (*action*, *subject*) to the *pool* according to the current dialog state. Every action can have an associated procedural *action rule* which is executed if the action is selected to be presented to the user.
3. *Select the most appropriate action* – the speech generation module (Section 5) selects the subset of the *pool* (actions which will be presented to the user) and executes the action rules associated with the selected actions. The action rule can add its own handler rules into the set of *handlers*. These handlers will be used the next turn.

The first step executes the *handler rules* used to process the observation. The handler rules ensure the processing of the observation in the context of the last dialog action. The handler rules most often handle confirmations, yes-no questions or requests. The *generation rules* are used to generate all possible actions in the current dialog state given the status of the slot or set of slots (for example request, confirm or disambiguate the slot value, present the result). If the required slots are filled also the task database is queried and the actions presenting the result to the user are inserted into the *pool*. Last the *action rules* are executed once the language generation module choses the actions which will be presented to the user.

The main advantage of the presented dialog manager is that it supports mixed initiative. The machine initiative is ensured by the generation step of the dialog strategy. The action (*request*:*slot*) is added into the pool if the given slot is not filled. If the action (*request*:*slot*) is selected to be present to the user, the machine takes the initiative and the user should react to this request. On the opposite side the user can take the dialog initiative by providing additional or different slot values in the observation. Also the *intention* attribute can change the state of the dialog manager.

4.3 Domain Specific Dialog Management

The current RIS-SDS uses five slots: *from_station* (representing the name of departing station, required slot), *to_station* (arriving station, required slot), *train_type* (the type of train the user wishes to use, optional slot), *time* (the time the user wishes to arrive or leave, optional slot), *time_is_arrival* (distinguishes the arrival/departure time). In addition the observation can contain the following values of the *intention* attribute: *greeting* (the user greeted the system), *next, previous* (the user wants the next or the previous train), *arrival, departure* (the user requests arrival or departure information), *back* (the return train is requested), *price, duration, transfer* (the user wants additional information about the train) and *other_info* (the out-of-domain utterance). During the development of RIS-SDS we also added a special slot *ref_time* which holds the reference time to support the correct interpretation of the time of day (a.m. and p.m.). The *ref_time* stores the last time used either by the user or by the system. The actions of the dialog manager includes actions managing the slots (*ask*, *disambig* or *confirm*), action for providing the database query result (*present*) and actions representing the polite phrases at the beginning and at the end of the dialog (*greeting* and *closing*).

5 Language Generation

The goal of the language generation module is to generate the textual representation of the actions in the *pool*. Because the *pool* can contain large number of possible actions (such as confirm a slot value, present a database result, disambiguate a slot value or request some required slot) only the subset of the *pool* can be presented to the user. Therefore the dialog manager works in three steps - the action rules are executed only if the corresponding action was chosen to be presented to the user. The language generation module uses the list of *templates* which maps the subset of the *pool* to the corresponding textual representation. An example of two templates can be:

```
{confirm:from_station, request:time} :
    "Which time you want to leave from $from_station?"
{confirm:from_station} :
    "Do you want to leave from $from_station?"
```

The most important feature of the language generation module is the possibility to join more than one action into one response. For example the first template in the example asks the user for the time he wants to leave and simultaneously the user can implicitly confirm the value of the slot *from_station*. If the slot *time* is already filled, the action (*request*:*time*) will not appear in the *pool* and the second template for the action (*confirm*:*from_station*) will be selected.

There can be many templates that matches the actions in the *pool* and the priorities of the templates have to be specified. The current implementation of RIS-SDS uses the position in the list of templates as the template priority. The templates at the beginning of the list are more specific and have higher priority than the less specific templates at the end of the template list. After selecting the template that match the *pool* and has the highest priority, the corresponding action rules are executed and the variables of the template are filled with the actual values. Than the text of the prompt is sent to the high-quality Czech text-to-speech module [8] and an acoustic response is presented to the user.

6 Conclusions and Future Work

The current version of the RIS-SDS is implemented as a desktop application and uses proprietary interface of the ASR and TTS modules. We plan to use the VoiceXML language to interact with the telephony infrastructure in the future RIS-SDS version.

The experiments with the prototype shown that the presented realization of the mixed-initiative spoken dialog systems provides a very good real-time response together with the natural style of a dialog. The dialog manager design is simple and the dialog manager rules can be easily modified and extended. In addition most of the rules can be shared between multiple domains. In the future experiments we want to focus on the collection of Human-Machine dialog corpora. Using these corpora we would be able to evaluate the performance not only of the whole system but also of the discrete modules of the dialog system (speech recognition, language understanding) in the real conditions. The Czech Human-Machine corpora will be also used to improve the dialog strategy by using statistical methods such as partially observable Markov decision processes (POMDPs). The design of the dialog state allows us to simply change the representation of some slots into the POMDP and integrate it into the generation step of the dialog strategy. Also the language generation module can use statistical methods to select the response according to the set of actions in the pool.

Acknowledgments

This work was supported by the Ministry of Education of the Czech Republic under project No. 1M0567 (CAK) and by the grant of The University of West Bohemia, project No. SGS-2010-054.

References

1. Young, S., Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K.: The Hidden Information State Model: A practical Framework for POMDP-Based Spoken Dialogue Management. Computer Speech and Language 24, 150–174 (2010)
2. Acomb, K., Bloom, J., Dayanidhi, K., Hunter, P., Krogh, P., Levin, E., Pieraccini, R.: Technical Support Dialog Systems: Issues, Problems, and Solutions. In: NAACL-HLT 2007: Proceedings of the Workshop on Bridging the Gap, Morristown, USA, pp. 25–31. ACL (2007)
3. Pražák, A., Psutka, J., Hoidekr, J., Kanis, J., Müller, L., Psutka, J.: Automatic Online Subtitling of the Czech Parliament Meetings. LNCS (LNAI), pp. 501–508. Springer, Heidelberg (2006)
4. Psutka, J.: Robust PLP-Based Parameterization for ASR Systems. In: SPECOM 2007 Proceedings. Moscow State Linguistic University, Moscow, pp. 509–515 (2007)
5. Jurčíček, F., Zahradil, J., Jelínek, L.: A Human-Human Train Timetable Dialogue Corpus. In: Proceedings of Interspeech, Lisboa, Portugal (2005)
6. Jurčíček, F., Švec, J., Müller, L.: Extension of HVS Semantic Parser by Allowing Left-Right Branching. In: Proc. IEEE ICASSP, Las Vegas, USA (2008)
7. Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., Mohri, M.: OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In: Holub, J., Žd'árek, J. (eds.) CIAA 2007. LNCS, vol. 4783, pp. 11–23. Springer, Heidelberg (2007)
8. Matoušek, J., Tihelka, D., Romportl, J.: Current State of Czech Text-to-Speech System ARTIC. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 439–446. Springer, Heidelberg (2006)

An NLP-Oriented Analysis of the Instant Messaging Discourse

Justyna Walkowska

Department of Computer Linguistics and Artificial Intelligence,
Faculty of Mathematics and Computer Science, Adam Mickiewicz University,
ul. Umultowska 87, 61-614 Poznań, Poland
justyna.walkowska@amu.edu.pl
http://www.amu.edu.pl/~zlisi

Abstract. This paper describes the results of the analysis of an experimentally collected small corpus of messages exchanged through an instant messaging (IM) programme. The data is analysed from the point of view of automatic parsing. Special attention is paid to two problems associated with IM discourse: the semantic multi-tasking (or the interweaving of topics) of conversation partners, and the non-standard spelling found in such dialogues. The contents of the corpus are also compared with other types of written dialogues, i.e. SMS messages and conversations between human users and chatterbots. Finally, some solutions are proposed to facilitate the process of automatic parsing of IM messages.

Keywords: machine dialogue, instant messaging, SMS messages.

1 Introduction

The last three years at Department of Computer Linguistics and Artificial Intelligence have been dedicated to the creation of Polint-112-SMS, a knowledge management system with natural language interface (SMS messages in Polish), designed to be used by security officers safeguarding mass events. The system [1] is now operational. In the design process we used a corpus-based methodology, described in [2]. The study of an SMS corpora [3] obtained early in the development process allowed us to draw a number of conclusions that strongly influenced the system's design and implementation.

This paper is dedicated to a form of speech in many ways similar to SMS messages, i.e. the messages sent by means of instant messaging (IM) programmes. Instant messaging is a type of online chatting in which users (normally two per dialogue) exchange messages through a client programme with people specified in a so called *buddy list*. The paper aims to present some important features of IM discourse and to discuss some problems that IM poses for automatic parsing and understanding.

2 Experimental Data

The main focus of this paper is experimental data collected over three days of February 2010. Ten pairs of people were asked to use an instant messenger (all chose Gadu-Gadu,

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 576–583, 2010.

very popular in Poland) to perform a dialogue. They were given two dialoguing goals. First, they had to choose a movie show to go to together. They were told that they did not need to actually go to the cinema, but that they should choose an existing showing that was convenient for both of them. They were allowed to use a web browser to check the cinema schedules and also were told that if their interlocutor took too much time responding, they were allowed to perform any online activity. The other goal of each of the conversation partners was to tell a short story (an anecdote, a news story).

The respondents were aged 20–30. The task was performed in average 22.8 minutes. The smallest number of messages sent by one person was 52, the largest was 102. The average number of messages exchanged during one minute was 6.86.

Some parts of this paper refer to a corpus of 3,000 Polish SMS messages described in [3], or to a small corpus of conversations between chatterbots and IT students performed during a Man-Machine Communication course.

3 IM Discourse

3.1 The Boundaries of Discourse

In [4] the authors name three distinctive features that individuate discourse: linguistic structure, intentions, and attention scope. All of the collected IM conversations may be cut down into smaller, coherent discourse segments, dedicated to the same topic. Their length varies from one message (short information not requiring any answer, e.g. *mam problemy z przeglądarką*, English *I'm having problems with my browser*) to as many as 35 messages (when telling a story).

Table 1. Changes of the focus of attention in one of the IM conversations

	Topic	**No. of messages**	**Continuity**
1.	Greetings	4	Continous
2.	Decision to go to the movies	6	Continous
3.	Possible movie: *Nine*	3	Continous
4.	Possible movie: *Christian Spirit/Lourdes*	10	Mixed with 5 and 6
5.	Apologies for using the Caps Lock key	1	Continous
6.	Possible movie: *It Might Get Loud*	6	Mixed with 4 and 7
7.	Possible movie: *Tatarak*	20	Mixed with 6
8.	Methods of searching for movies	4	Mixed with 9
9.	Possible movie: *Dom zły*	13	Mixed with 8
10.	Possible movie: *Sherlock Holmes*	9	Continous
11.	Possible movie: *Millennium*	17	Continous
12.	Time of the showing	16	Continous
13.	Anecdote 1	14	Continous
14.	Anecdote 2	15	Mixed with 15
15.	Trying to end the conversation	6	Mixed with 14
16.	Goodbyes	4	Continous

Table 1 presents the discourse pieces in one of the IM conversations. It is clearly visible that in almost half of the cases discourse segments are overlapping or interwoven.

The same thing occurred in all of the experimental IM conversations, but not always to this extent. It seems that people who use instant messengers more often tend to talk about more than one subject at once. Multitasking has been mentioned in [5], but in the context of using other programmes, mostly web browsers and e-mail clients, while talking through instant messengers, and not in the context of talking about multiple subjects at once.

In the recent years there has been a lot of discussion about multitasking vs attention span, especially among students who seem to no longer be able to focus on one subject for the expected amount of time. The blame for this state of affairs is often placed on human interaction with computers. However, in the case of IM dialogues, it seems that the conversation partners do not deliberately choose to multitask. Multitasking and discourse-interweaving (and also separation of questions and their respective answers within one topic) happen when:

- Person A types a message, missing (not reading) some messages written by their interlocutor B. The most common behaviour for A is then to react to the newest message first, and check the other unread messages later.
- A sends a sequence of messages and misses a message (usually a question) sent by B. A answers the question at a later time.
- A needs time to type a question while B is sending more messages.
- A needs time to look something up (e.g. a cinema schedule) online and B starts a new subject while waiting.
- A conversation topic is exhausted and two people start talking about new things (possibly asking questions) at once. In a spoken conversation one person would probably apologize for cutting in and only one subject would be discussed at the time.

Below is a short piece of conversation summarized in Table 1 (end of segment 14).

(1) (09:07:45) A: siedzialam przed kompem nowego Burzum sluchajac (*I sat at my PC listening to the new Burzum*)
(2) (09:07:51) A: heheheh (3) (09:07:59) B: dobry album? (*is the album any good?*)
(4) (09:08:04) A: dobra (*ok*)
(5) (09:08:04) A: pogadamy wieczorem (*we'll talk in the evening*)
(6) (09:08:10) B: tez musze wychodzic (*I have to go too*)
(7) (09:08:13) B: bez odbioru (*over and out*)
(8) (09:08:19) A: album epicki (*the album is epic*)

In lines (1) and (2) person A finishes to tell an anecdote about a night spent at home. In line (3) B asks a question (which they probably started writing after reading (1)). At the same time A starts a new subject (trying to finish the conversation, (4) and (5)), to which B reacts in (6) and (7). A finally reads the question asked a few messages ago and answers it in (8).

3.2 Comparison with SMS Messages

The extent to which the topics in IM conversations interweave may seem scary from the point of view of automatic understanding. Let us compare them with some other forms of written communication.

The analysis of the aforementioned corpus of Polish SMS messages showed that SMS and IM discourse bear strong resemblance in some aspects. In both modes, when the exchange of messages is intensive, users often prefer to finish typing an already started message before looking up any new incoming information. In the case of SMS this tendency is even stronger, because to read the incoming message the phone user would have to close the one they are already typing, which could result in losing the unfinished text. Because of it in both cases the resulting discourse is often inconsistent.

However, there are differences. Firstly, typing and sending an SMS message is more costly (in time, effort, and money), so SMS messages tend to be longer: in particular, sentences are never cut between messages, which in IM dialogues happens a lot. This, of course, limits the interweaving. Secondly, in IM conversations it is easier to check the dialogue history. In consequence, IM users sometimes answer questions asked many (the maximum in the experimental corpus is 20) messages ago. If SMS users miss any messages, or open them too late (after exchanging other messages) they are less likely to try to answer them, because often they are no longer able to place them in a context.

3.3 Can IM Discourse Be Parsed?

As stated in the introduction, the original research on SMS messages (and partly also IM messages) was conducted for the purposes of creating a knowledge-management system with natural language interface. After an initial glance at the results one may think that it is impossible for a chatterbot-like computer system to cope with a discourse so untidy. However, there are reasons for optimism.

A study of a small corpus of students' conversations with chatterbots held in Polish, English or German, showed that when people are aware that they are talking to a machine, they switch to a less demanding conversation mode. Even when students tried to purposely confuse the bots (especially those programmed by their fellow students) they usually did so lexically, using words they hoped the bots would not understand. In almost all cases the conversation was comprised of subsequent question-answer pairs; the students did not expect the chatterbots to be able to participate in a normal, multi-topic information exchange.

Still, if humans are able to follow the discourse and recognize which messages belong to which segment, there must be markers that have to be properly recognized. In the collected corpora, every time when there are two active, interwoven topics, they are either very different (e.g. person/movie), so in particular the pronouns may be interpreted easily, or the speakers become aware of the problem and start using fuller sentences. In the conversation extract above there are as many as 5 messages between the question about the album and the answer. The word *album* is thus repeated to make the answer unambiguous.

It does happen that two people engaged in an IM conversation finally get lost among too many topics. What they do then is the simplest: they ask for clarification, and it is also what a language-competent virtual agent should do. Below is an example from the IM corpus.

A: dzis o 17 w malcie jest (*it is shown in malta (cinema) today at 5pm*)
B: co takiego? (*what is?*) A: dom zly (*dom zly)*)

4 Spelling and Correction

In most papers dedicated to the description and analysis of SMS or instant messages, a lot of stress is placed on the spelling ([6,5]). The spelling does differ from what can be found in more formal texts, which seems to cause concern among the researchers, both from the language purity and automatic understanding points of view. [7] describes a corpus of 30,000 SMS messages in French that was "translated" to standard French manually.

Having analysed the corpora gathered for the Polint-112-SMS system [3], and the newly obtained IM corpora, we categorized the most common deviations from correct (or standard) Polish language, and proposed general solutions to facilitate the process of automatic parsing. The detected categories are:

- typographical errors (typos),
- spelling mistakes,
- T9 errors (SMS only),
- beginnings of words instead of full words,
- acronyms,
- words spelled without vowels,
- phonetic spelling,
- CamelCase (SMS only),
- emoticons,
- other.

Typographical Errors. Typos are more common in SMS and IM messages than in formal texts because of the very quick nature of these types of communication. Over the years many methods of correction have been proposed, the most famous being the Levenshtein algorithm [8]. The algorithm is based on so-called *edit distance* metric. The distance between two words is defined as the minimum number of edit operations needed to transform one string into the other. The operations are: insertion, deletion, and substitution of a single character.

Some typos are more common than others. In SMS messages, when users type on the limited keyboard with a number of letters assigned to one key (letters *a*,*b*,*c* under the *2* key, in Polish also two additional letters with diacritic signs) it is probable that sometimes a wrong letter appears in the text. While using a regular *qwerty* keyboard (in IM messages, but also in SMS messages written on *smartphones* with extended keyboards) it is probable that the user will, from time to time, hit a key adjacent to the one they intended to hit.

A known solution for this type of errors the Levenshtein algorithm with weights. The more common substitutions (e.g. *a* and *b* at a phone keyboard) receive lower weight values (e.g. 0.1 instead of 1), so the resulting edit distance between the word with such errors and the correct version (from a dictionary) is very small. Words whose similarity is smaller than a threshold value (often depending on the length of the words) may be interchanged. This algorithm can also be used to add diacritical signs and accents.

It is worth noting that sometimes the wrongly spelled word may in fact be a correct word with different meaning (English *miss* and *kiss*). To automatically detect this kind of errors it is necessary to involve an understanding module that is able to exclude existing words that do not make sense syntactically or come from beyond the system's domain.

Spelling Mistakes. Although different in nature from typographical errors (they result from the user's ignorance and not a technical mistake), spelling mistakes can be corrected using the same methods. In the original edit distance metric the substitution only concerns single characters, but the algorithm may be extended. The "traditional" spelling errors (homophones) in Polish are *ż* vs *rz*, *u* vs *ó*, and *h* vs *ch*, so to correct them it is sufficient to extend the algorithm by allowing multi-letter substitutions and adjusting weights.

T9 Errors. T9 errors only pose a problem for SMS communication. The T9 dictionary application makes it possible for the phone user to type a word pressing a key only once for each letter, even if the letter normally is not the first one for the key. For example, the Polish word *dom* (*house*) may be typed with the keys 3-6-6 and each one of them is pressed only once. Unfortunately, often more then one word can be spelled with the same sequence of keys. Another example for 3-6-6 may be the word *dno* (*bottom*). Often the users, accustomed to the fact that normally the expected word appears first, do not check if the result is correct. Again, a variation of the Levenshtein algorithm may be applied, where the T9 substitutions are the only allowed ones. The case is more difficult than simple typos because the incorrect strings are always existing words.

Beginnings of Words. The phenomenon of using only the first couple of letters of a word is present in corpora of different languages. Some shortened words are introduced into official, standard language (e.g. the English word *typo*). In most cases the original can be found by searching for words starting with the given substring. It is possible, of course, that more than one word will be found (or different syntactic forms of the same word). Again, and additional module would be necessary to exclude words based on the syntax and domain.

Acronyms. There is a number of IM, SMS and e-mail related acronyms. Some are international (as *ROTFL*), some not (Polish *ZW*, *Zaraz Wracam*, i.e. *i'll back in a minute*). It is important to realize that normally humans do not understand those acronyms when they see them for the first time – most have to ask for explanation. The number of those acronyms is limited (even if constantly growing), so the proposed solution is to add them to the dictionary. Experimental results [3] showed that people usually do not use the acronyms when they are aware that they are talking to a computer system.

Words Spelled Without Vowels. Both in Polish and English, vowels are more redundant than consonants, so IM or SMS users who want to save time sometimes decide to resign from using them. It is easy to recognize words that were subject to this process (they do not contain vowels, or only contain one vowel if the word starts with it). To find the original word it is necessary to search the dictionary, possibly using a variation of the omnipresent Levenshtein algorithm in which vowel insertion comes at almost no cost.

Phonetic Spelling. Phonetic spelling (e.g. *q* instead of Spanish *que*) is present in languages in which spelling and pronunciation are not tightly bound. In Polish it happens rarely. Few examples were noted in the corpora: the number *3* (Polish *trzy*) was used as a part of a word (*3mam* as *trzymam*), and twice the spelling of a word was changed to resemble English. Example (from beyond the corpus): the Polish word *kultura* (*culture*) by some people is spelled *cooltura*. Those cases are rare. Some are quite hopeless, for some one may try to introduce a smaller substitution weight (e.g. between *3* and *trzy*).

CamelCase. CamelCase, i.e. writing compound words or phrases without spaces, with the first letter of each word capitalized, was popular when SMS messages were expensive to send, so the users tried to squeeze more information in one message. It is less popular today (2 messages in the SMS corpus), also because typing a message in this style requires more effort. This type of spelling is easy to recognize and correct by splitting the string at capital letter positions.

Emoticons. It is debatable whether emoticons provide information that should be analysed by automatic parsers. If one decides not to pay attention to them, they have to be removed carefully, as simple deletion may cause problems. The punctuation (or lack thereof) of IM and SMS messages is often limited. Emoticons sometimes play the role of punctuation marks, as in the following example:

> Person A: znasz go? (*do you know him?*)
> Person B: nie :) znam jego brata (*no :) I know his brother*)

Failing to replace the emoticon with a punctuation mark (a coma or a full stop) changes the Polish sentence to *nie znam jego brata* (*I don't know his brother*).

Other. The categories from this section summarize most of the problems (and proposed solutions) with unorthodox spelling of short messages. There are two other phenomena in the corpora which have to do with spelling. Both are rare. The first one is the presence of foreign words (mostly English and French) in Polish messages. The other is the existence of word variations invented by individuals, that are still understood by their partners (e.g. *sorkacz* which originates from the English word *sorry*). The automatic normalization of those cases would be more difficult and computationally heavier.

5 Conclusions

In this paper two important and “problematic” features of IM communication have been discussed: the interweaving of discourse topics caused by the users’ multitasking, and the non-standard spelling of the messages. A number of solutions have been proposed for the described challenges, most of which have been successfully applied in the Polint-112-SMS application. Hopefully they can facilitate the development of other systems with NL competence.

References

1. Vetulani, Z., Marciniak, J., Konieczka, P., Walkowska, J.: An SMS-Based System Architecture (Logical Model) To Support Management of Information Exchange in Emergency Situations. POLINT-112-SMS project. In: Intelligent Information Processing IV. 5th IFIP International Conference on Intelligent Information Processing, pp. 240–253. Springer, Boston (2008)
2. Vetulani, Z., Marciniak, J.: Corpus Based Methodology in the Study and Design of Systems with Emulated Linguistic Competence. In: Christodoulakis, D.N. (ed.) NLP 2000. LNCS (LNAI), vol. 1835, pp. 346–357. Springer, Heidelberg (2000)
3. Walkowska, J.: Gathering and Analysis of a Corpus of Polish SMS Dialogues. In: Challenging Problems of Science. Computer Science. Recent Advances in Intelligent Information Systems, pp. 145–157. Academic Publishing House EXIT, Warsaw (2009)
4. Grosz, B.J., Sidner, C.L.: Attention, Intentions and the Structure of Discourse. Computational Linguistics 12(3), 175–204 (1986)
5. Hult, C.A., Richins, R.: The Rhetoric And Discourse Of Instant Messaging. Computers and Composition Online (2006), http://www.bgsu.edu/cconline/hultrichins_im/hultrichins_im.htm
6. Thurlow, C., Brown, A.: Generation Txt? The Sociolinguistics of Young People’s Text-Messaging. Discourse Analysis Online (2003), http://extra.shu.ac.uk/daol/articles/v1/n1/a3/thurlow2002003-01.html
7. Fairon, C., Paumier, S.: A Translated Corpus of 30,000 French SMS. In: Proceedings of LREC 2006, Genoa (2006)
8. Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady 10, 707–710 (1966)

Expressive Gibberish Speech Synthesis for Affective Human-Computer Interaction

Selma Yilmazyildiz, Lukas Latacz, Wesley Mattheyses, and Werner Verhelst

Interdisciplinary Institute for Broadband Technology – IBBT
Vrije Universiteit Brussel, dept. ETRO, Belgium
{syilmazy,llatacz,wmatthey,wverhels}@etro.vub.ac.be
http://www.ibbt.be, http://www.etro.vub.ac.be

Abstract. In this paper we present our study on expressive gibberish speech synthesis as a means for affective communication between computing devices, such as a robot or an avatar, and their users. Gibberish speech consists of vocalizations of meaningless strings of speech sounds and is sometimes used by performing artists to express intended (and often exaggerated) emotions and affect, such as anger and surprise, without actually pronouncing any understandable word. The advantage of gibberish in affective computing lies with the fact that no understandable text has to be pronounced and that only affect is conveyed. This can be used to test the effectiveness of affective prosodic strategies, for example, but it can also be applied in actual systems.

Keywords: Affective speech synthesis, expressive speech synthesis, gibberish speech.

1 Introduction

The desire of mankind for an intelligent interaction with machines (HCI) has been among the mostly dreamed of concepts in science fiction and exact science alike. Today, humans communicate with machines in everyday life. Navigation systems, car diagnosis systems, computer games, distance learning applications, assistive technologies and robotic assistants are just a few examples. Day by day, HCI resembles more and more the natural interaction between humans (HHI). One of the most important interaction features of HHI is expressive speech communication, which allows the communication of affect and intent and this not only between humans but also with animals, as used in animal assisted therapy [1] and robot assisted therapy [2].

In HHI, only 7% of information is transferred by the words spoken while 38% is transferred by the tone of voice [3]. Therefore, a nonsense language like gibberish could be successful as a carrier to express emotions and affect. Gibberish could even be more advantageous than plain speech since no understandable text has to be pronounced and the focus is only on conveyed affect. Our aim is to contribute to affective HCI a non-understandable/gibberish language and to build expressively interacting computing devices. We also intend to experiment with affective gibberish speech for communication between robots and children.

The paper is organized as follows: in Section 2 we describe our approach for gibberish text generation; in Section 3 we discuss its usage as a front end for text

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 584–590, 2010.

to speech synthesizers (TTS); in Section 4 we investigate the correlations between perceived and intended emotions in both plain and gibberish speech and in Section 5 we conclude with a discussion.

2 Gibberish Speech Synthesis

Siblings sometimes use toy language. Such language can be either a coded form of their mother tongue (e.g., “mother tongue” becomes “mopotheper topongue” in the p-language) or can be meaningless (i.e., gibberish). Meaningless speech can also be used as a segmental evaluation method for synthetic speech [4], to test the effectiveness of affective prosodic strategies [5], for example, but it can also be applied in actual systems [6,7].

Languages consist of ruled combinations of words and words consist of specially ordered syllables. Syllables are often considered the phonological “building blocks” of the words and they usually contain an “onset”, a “nucleus” and a “coda”. In most languages, every syllable requires a nucleus which is usually a vowel-like sound. In English and Dutch, vowel nuclei are transcribed with one, two and three letters. There are usually only a few vowel nuclei with one letter transcriptions but they are most frequently used in the language (Fig. 1). There are usually much more vowel nuclei with two or three letter transcriptions, but these are far more rarely used.

To produce gibberish speech, we wrote a program that replaces the vowel nuclei in a text with other vowel nuclei of the same language such that the text looses its meaning. We then used that gibberish text as input for TTS engines to generate the gibberish speech. However, if we would transform the word “language” into gibberish with a uniform random swapping of vowel nuclei, we would likely end up with something like “lieungeaugie”. To avoid this, we calculated the probabilities of occurrence for each vowel nucleus and used a weighted swapping mechanism in accordance with the probabilities instead of uniform random swapping. Fig. 1 represents the empirical probability distributions of vowel nuclei for English and Dutch text of approximately 27,000 words each from Project Gutenberg [8].

3 Input Text and Language

Our goal being to create a *natural sounding* gibberish language, we transform existing text into meaningless text and use this as input text for a TTS. However, as the TTS’s language processing modules are not designed to work on meaningless text we investigated how natural our synthetic gibberish sounds and whether the native language of the TTS affects the result.

We therefore created two sets of sentences. For the first set, 6 original English sentences were selected from children’s stories [8] and converted to gibberish using the English vowel nuclei probability distributions. For the second set, 6 sentences were selected from Dutch children’s stories [8] and converted to gibberish using the Dutch vowel nuclei probability distributions. We then synthesized all 12 gibberish sentences both with the Dutch and the English version of our unit selection TTS [9]. We thus constructed 4 different groups of samples: 6 samples with Dutch gibberish text and

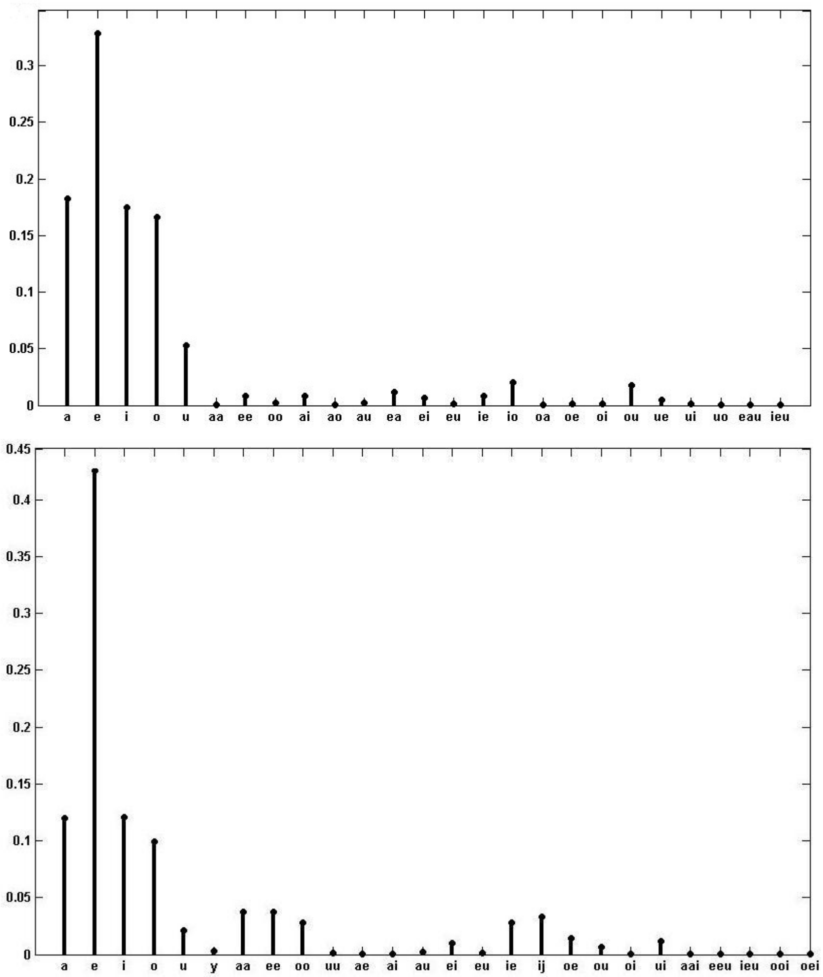


Fig. 1. Empirical probability mass distribution of vowel nuclei in English (upper panel) and Dutch (lower panel)

Dutch TTS, 6 samples with English gibberish text and English TTS, 6 samples with Dutch gibberish text and English TTS, and 6 samples with English gibberish text and Dutch TTS.

Ten subjects aged between 24 and 37 participated in a listening experiment. Four subjects had no prior experience with synthetic speech. The subjects were asked to pay attention to the naturalness of the samples. They were instructed that a sample is to be considered as natural when it sounds more like an unrecognized real language than like an unnatural or random combination of sounds. They were asked to express their judgement using Mean Opinion Scores (MOS) on a scale of 1 to 5. We also asked them to write down the language if the sample sounded like a language they knew. For the naïve subjects, we provided an example of natural (i.e., plain) synthetic speech at the beginning of the test to ensure that they would not rate the quality of the TTS instead of the naturalness of the gibberish.

Table 1 shows the average MOS and the results of the 2 × 2 ANOVA on the MOS scores. The two variables are the original language of the input gibberish text (TEXT) and the language used for synthesis (SYNTH). The gibberish speech was perceived as natural by most of the subjects with an overall MOS of 3.62. The language of the synthesizer had a significant influence (Sig. = 0.043) on the perceived naturalness but no significant influence of the input language and no combined effect were found. The samples created with the Dutch synthesizer had the highest score for both Dutch and English gibberish texts. That could be because almost half of the subjects were native Dutch speakers. Informal listening has shown that the Dutch synthesizer has better quality than the English version, which could be another possible explanation. In general, we can conclude that all versions of the gibberish speech synthesizer were found to be rather natural sounding.

Table 1. Experimental results and statistical analysis (sig. threshold level α=0.05)

Test Results		
TEXT	SYNTH	Mean MOS
Dutch	Dutch	3.82
Dutch	English	3.42
English	Dutch	3.68
English	English	3.55
General Mean		3.62

2 × 2 ANOVA			
Factor	df	*F*	Sign.
TEXT	1	0.000	1.000
SYNTH	1	4.137	0.043
TEXT*SYNTH	1	1.034	0.310

Fig. 2 shows to what extent the subjects were able to identify the original language in the gibberish samples. It is seen that for both Dutch and English, the recognition rates are highest when both the gibberish input text language and the synthesizer language are the same. In general the synthesizer language has more impact while the text language has only little effect. Dutch was more easy to recognize with scores up to 78% while for English the highest recognition rate was about 50%. This is most likely due to the fact that almost half of the subjects were native Dutch speakers and that the Dutch synthesizer has better quality than the English version. An important conclusion is that gibberish speech with the correct probability distribution of Dutch vowel nuclei and synthesized with a Dutch TTS system does indeed resemble Dutch. Together with the results presented in Table 1, we may thus conclude that we achieved our goal of constructing a gibberish synthesizer that sounds like natural Dutch.

4 Semantic Meaning and Perceived Emotion

People naturally use both prosodic meaning and semantic meaning for expressing affect and emotion. In gibberish, there is no semantic information. Furthermore, the fact that gibberish is meaningless might interfere with the prosodic strategy of the synthesizer

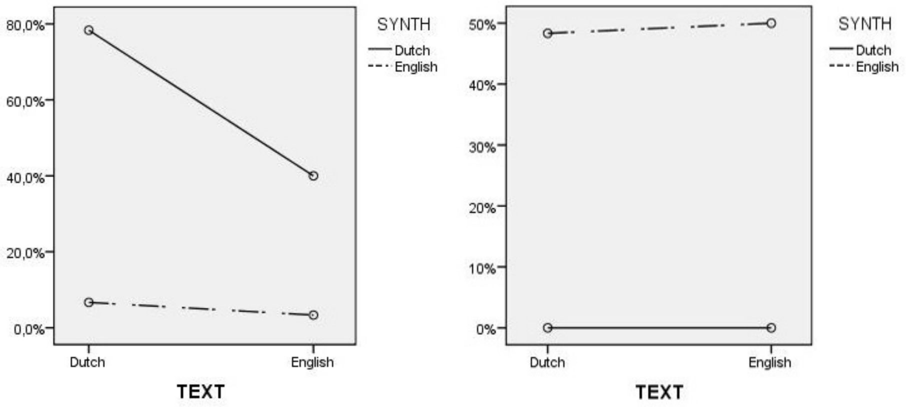


Fig. 2. Percentages of language recognition for Dutch (left panel) and English (right panel)

and result in less expressive speech. Therefore, we investigated whether the semantics of the underlying text influence the perception of emotions in synthetic speech and whether gibberish might be more or less effective than plain speech in conveying the intended emotion.

We synthesized 4 groups of samples. In the first group, the semantic meanings of the sentences and the acoustic properties of the synthesized utterances correspond to the same emotion. In the second group, the semantic meanings of the sentences have opposite emotion of the acoustic properties. In the third group, the semantic meanings of the sentences are neutral, and in the fourth group the sentences are gibberish and therefore have no semantic meaning. Each group contains 10 samples. The emotion categories used were happy and sad. We used the open source emotional TTS synthesis

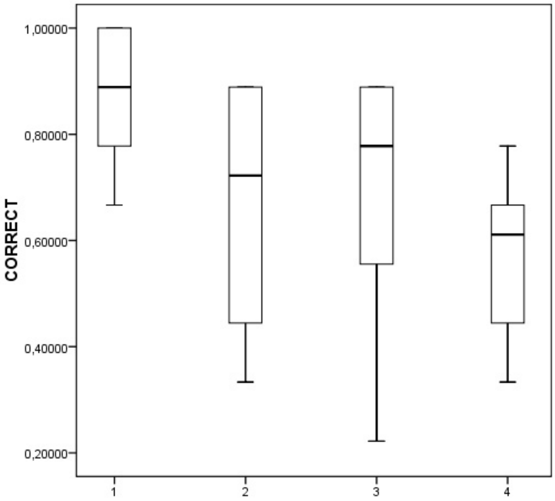


Fig. 3. Box plot of the emotion recognition results for 4 different experimental groups

tool, “EmoSpeak”, of the synthesizer Mary [10,11] with the parameter settings for happy and sad reported in [12] to produce the emotional speech.

Nine subjects aged between 26 and 37 joined the forced-choice listening test. Three subjects had no experience with synthetic speech. The subject were instructed to listen to a number of samples of which they may or may not understand the meaning and they were requested to choose which one of the possible emotions happy, sad or neutral matches the sample they heard.

Fig. 3 shows the emotion recognition results for all 4 groups. Group 1 (semantic meaning and acoustics correspond to the same emotion) has the highest scores among all groups. The other 3 groups showed comparable recognition results amongst each other. A Kruskal-Wallis test confirmed that Group 1 is indeed significantly different (Sig.=0.032). Thus, semantic meaning did help for recognizing the intended emotion, as expected. On the other hand, semantics opposite to the intended emotion did not make the task more difficult than with neutral semantics or with gibberish speech.

5 Discussion

In the first experiments we explored the influence of input text and language on the synthesized gibberish speech. It was found that gibberish speech resembles natural language with a total average MOS of 3.62. We did find a significant difference between naïve subjects and subjects having expertise with synthetic speech. The overall naturalness ratings of the naïve subjects were significantly lower than the ratings of the speech experts. We received feedback from the naïve subjects that they found it difficult to evaluate the naturalness of gibberish speech independent from the synthesis quality; they believe that the synthetic speech quality may have negatively influenced their scores. Also, the samples from the Dutch synthesizer received higher scores than the samples from its English version, which is likely due to the difference in quality between the Dutch and English versions of the synthesizer as the Dutch synthesis database is almost 4 times larger than the English database.

The experiments also showed that the gibberish language resembles the source language when a good quality synthesis is used in combination with an input text from the same language. This can be easily understood as, even without semantic meaning, the synthesizer still uses the phones and intonation model of its target language. Moreover, the gibberish input text for the synthesizer has the same vowel distribution as the source language.

From the experiments on the relation between semantic meaning and the perceived emotion, we found that semantics help for recognizing the intended emotions when the semantic and the prosodic meaning of the utterances are both in line with the intended emotion. When they were in line with opposite emotions, this did confuse the subjects but less so than might have been expected. A probable cause is that the synthesizer we used simulates happy with high speaking rate and sad with low speaking rate such that the intended emotion could be easily inferred. We received feedback from subjects that they did indeed mostly use speaking rate as a clue to infer the intended emotion.

No statistical difference was found between samples with emotionally neutral meaning and gibberish samples. As a consequence, we can say that gibberish speech conveys the emotions as effectively as semantically neutral speech and can be used in an affective communication system. It should be noted, however, that this conclusion is valid for our experiments on emotional speech synthesis with the Mary synthesizer and it is not at present clear to what extent similar results would be obtained with other expressive synthesizers.

Acknowledgments. The research reported on in this paper was supported in part by the Research counsel of the Vrije Universiteit Brussel with horizontale onderzoeksactie HOA16, by the Flemish government (IBBT project SEGA) and by the European Commission (EU-FP7 project ALIZ-E).

References

1. Heimlich, K.: Animal-Assisted Therapy and the Severely Disabled Child: A Quantitative Study. Journal of Rehabilitation 67(4), 48–54 (2001)
2. Shibata, T., Mitsui, T., Wada, K., Touda, A., Kumasaka, T., Tagami, K., Tanie, K.: Mental Commit Robot and Its Application to Therapy of Children. In: Proc. of the IEEE/ASME International Conference on AIM 2001, p. 182 (July 2001)
3. Mehrabian, A.: Communication Without Words. Psychology Today 2(4), 53–56 (1968)
4. Carlson, R., Granström, B., Nord, I.: Segmental Evaluation Using the Esprit/SAM Test Procedures and Mono-syllabic Words. In: Bailly, G., Benont, C. (eds.) Talking Machines, pp. 443–453. Elsevier, North Holland (1990)
5. Yilmazyildiz, S., Mattheyses, W., Patsis, G., Verhelst, W.: Expressive Speech Recognition and Synthesis as Enabling Technologies for Affective Robot-Child Communication. In: Zhuang, Y., Yang, S., Rui, Y., He, Q. (eds.) PCM 2006. LNCS, vol. 4261, pp. 1–8. Springer, Heidelberg (2006)
6. Oudeyer, P.Y.: The Synthesis of Cartoon Emotional Speech. In: Proc. of the 1st International Conference on Prosody, Aix-en-Provence, France, pp. 551–554 (2002)
7. Breazal, C.: Sociable Machines: Expressive Social Exchanges Between Humans and Robots. Ph.D. thesis, MIT AI Lab (2000)
8. Hart, M.: Project Gutenberg (2003), http://www.gutenberg.org
9. Latacz, L., Kong, Y., Mattheyses, W., Verhelst, W.: An Overview of the VUB Entry for the 2008 Blizzard Challenge. Blizzard Challenge 2008, In: Brisbane, Australia (2008)
10. Schröder, M.: Speech and Emotion Research: An Overview of Research Frameworks and Dimensional Approach to Emotional Speech Synthesis. Ph.D. thesis, PHONUS 7, Research Report of the Institute of Phonetics, Saarland University (2004)
11. OpenMary: Open Source Emotional Text-to-Speech Synthesis System, http://mary.dfki.de/
12. Schröder, M., Cowie, R., Douglas-Cowie, E., Westerdijk, M., Gielen, S.: Acoustic Correlates of Emotion Dimensions in View of Speech Synthesis. In: Proc. of the Eurospeech 2001, Aalborg vol. 1, pp. 87–90 (2001)

Author Index