Ivan Habernal
Václav Matoušek (Eds.)

LNAI 6836

# Text, Speech and Dialogue

14th International Conference, TSD 2011
Pilsen, Czech Republic, September 2011
Proceedings

T S D

Springer

# Lecture Notes in Artificial Intelligence 6836

## Subseries of Lecture Notes in Computer Science

Ivan Habernal   Václav Matoušek (Eds.)

# Text, Speech and Dialogue

14th International Conference, TSD 2011
Pilsen, Czech Republic, September 1-5, 2011
Proceedings

Springer

Volume Editors

Ivan Habernal
Václav Matoušek
University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Univerzitní 8, 306 14 Plzeň, Czech Republic
E-mail: {habernal, mtousek}@kiv.zcu.cz

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-23537-5 e-ISBN 978-3-642-23538-2
DOI 10.1007/978-3-642-23538-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011934757

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

*Typesetting:* Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

TSD 2011 was the 14th event in the series of International Conferences on Text, Speech and Dialogue supported by the International Speech Communication Association (ISCA) and the Czech Society for Cybernetics and Informatics (ČSKI). This year, TSD was held in Plzeň (Pilsen), in the new Conference Center of the Vienna International Hotel Angelo, during September 1–5, 2011 and it was organized by the University of West Bohemia in Plzeň in cooperation with Masaryk University of Brno, Czech Republic. Like its predecessors, TSD 2011 highlighted to both the academic and scientific world the importance of text and speech processing and its most recent breakthroughs in current applications. Both experienced researchers and professionals as well as newcomers to the text and speech processing field, interested in designing or evaluating interactive software, developing new interaction technologies, or investigating overarching theories of text and speech processing, found in the TSD conference a forum to communicate with people sharing similar interests. The conference is an interdisciplinary forum, intertwining research in speech and language processing with its applications in everyday practice. We feel that the mixture of different approaches and applications offered a great opportunity to get acquainted with current activities in all aspects of language communication and to witness the amazing vitality of researchers from developing countries too.

This year's conference was partially oriented toward integrating the modern Web with speech and language technologies, which was chosen as the main topic of the conference. All invited speakers – Hynek Hermansky (Johns Hopkins University, Baltimore), Tatiana Skrebtcova (Saint-Petersburg State University), and Mark Epstein (Google, New York City) – gave very nice invited talks on the latest results in the relatively broad and still unexplored area of integrating Web and speech technologies. Many interesting questions of the newest speech applications for Web communication were answered in these talks; the convergence between Western and Eastern countries in realization of Web applications can be seen as one of the most significant results of the conference.

This volume contains a collection of submitted papers presented at the conference, which were thoroughly reviewed by three members of the conference reviewing team consisting of more than 60 top specialists in the conference topic areas. A total of 53 accepted papers out of almost 110 submitted, altogether contributed by about 140 authors and co-authors, were selected by the Program Committee for presentation at the conference and for inclusion in this book. Theoretical and more general contributions were presented in common (plenary) sessions. Problem-oriented sessions as well as panel discussions then brought together specialists in limited problem areas with the aim of exchanging knowledge and skills resulting from research projects of all kinds. Finally, the student session could be seen as the newest part of the conference. It offered eight selected PhD students the chance to discuss some problems and results.

The organization of the Third International Workshop on Balto-Slavonic Natural Language Processing affiliated to the TSD 2011 Conference was the second significant novelty of the conference program. Eight strictly selected speakers presented very interesting comparisons of machine processing of Slavonic natural languages used in Eastern Europe and they contributed with their presentations to the success of this year's conference.

All conference participants missed our long-time Program Committee Chairman Frederick Jelinek, who passed away immediately after the last TSD conference in Brno. May his soul rest in peace.

Last but not least, we would like to express our gratitude to the authors for providing their papers on time, to the members of the conference reviewing team and Program Committee for their careful reviews and paper selection and to the editors for their hard work preparing this volume. Special thanks are due to the members of the Local Organizing Committee for their tireless effort and enthusiasm during the conference organization.

June 2011 Václav Matoušek

# Organization

TSD 2011 was organized by the Faculty of Applied Sciences, University of West Bohemia in Plzeň (Pilsen), in cooperation with the Faculty of Informatics, Masaryk University in Brno, Czech Republic. The conference website is located at:

`http://www.kiv.zcu.cz/tsd2011/` or `http://www.tsdconference.org` .

## Program Committee

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

## Local Organizing Committee

Václav Matoušek *(Chair)*
Tomáš Brychcín
Kamil Ekštein
Ivan Habernal
Jan Hejtmánek
Michal Konkol
Miloslav Konopík
Pavel Mautner
Roman Mouček
Jana Vlčková *(Secretary)*

## Sponsoring Institutions

International Speech Communication Association (ISCA)
Czech Society for Cybernetics and Informatics (CSKI)

# About Plzeň (Pilsen)

The new town of Pilsen was founded at the confluence of four rivers – Radbuza, Mže, Úhlava and Úslava – following a decree issued by the Czech king, Wenceslas II. He did so in 1295. From the very beginning, the town was a busy trade center located at the crossroads of two important trade routes. These linked the Czech lands with the German cities of Nuremberg and Regensburg.

In the fourteenth century, Pilsen was the third largest town after Prague and Kutna Hora. It comprised 290 houses on an area of 20 ha. Its population was 3,000 inhabitants. In the sixteenth century, after several fires that damaged the inner center of the town, Italian architects and builders contributed significantly to the changing character of the city. The most renowned among them was Giovanni de Statia. The Holy Roman Emperor, the Czech king Rudolf II, resided in Pilsen twice between 1599 and 1600. It was at the time of the Estates revolt. He fell in love with the city and even bought two houses neighboring the town hall and had them reconstructed according to his taste.

Later, in 1618, Pilsen was besieged and captured by Count Mansfeld's army. Many Baroque-style buildings dating to the end of the seventeenth century were designed by Jakub Auguston. Sculptures were made by Kristian Widman. The historical heart of the city – almost identical to the original Gothic layout – was declared a protected historic city reserve in 1989.

Pilsen experienced a tremendous growth in the first half of the nineteenth century. The City Brewery was founded in 1842 and the Skoda Works in 1859. With a population of 175,038 inhabitants, Pilsen prides itself on being the seat of the University of West Bohemia and Bishopric.

The historical core of the city of Pilsen is limited by the line of the former town fortification walls. These gave way, in the middle of the nineteenth century, to a green belt of town parks. Entering the grounds of the historical center, you walk through streets that still respect the original Gothic urban layout, i.e., the unique developed chess ground plan.

You will certainly admire the architectonic dominant features of the city. These are mainly the Church of St. Bartholomew, the loftiness of which is accentuated by its slim church spire. The spire was reconstructed into its modern shape after a fire in 1835, when it was hit by a lightning bolt during a night storm.

The placement of the church within the grounds of the city square was also rather unique for its time. The church stands to the right of the city hall. The latter is a Renaissance building decorated with graffiti in 1908–12. You will certainly also notice the Baroque spire of the Franciscan monastery.

All architecture lovers can also find more hidden jewels, objects appreciated for their artistic and historic value. These are burgher houses built by our ancestors in the styles of the Gothic, Renaissance or Baroque periods. The architecture of these sights was successfully modeled by the construction whirl of the end of the nineteenth century and the beginning of the twentieth century.

Thanks to the generosity of the Gothic builders, the town of Pilsen was predestined for free architectonic development since its very coming to existence. The town has therefore become an example of a harmonious coexistence of architecture both historical and historicizing.

# Table of Contents

## Invited Talks

## Conference Papers

## Balto-Slavonic Natural Language Processing 2011 Workshop

# Dealing with Unexpected Words in Automatic Recognition of Speech

Hynek Hermansky[1,2]

[1] Center for Language and Speech Processing
The Johns Hopkins University
Baltimore, Maryland, USA
[2] Brno University of Technology
Czech Republic

**Abstract.** Unexpected words attract listener's attention. They are information-rich and getting them right is important for human communication. In the automatic recognition of speech (ASR), words that are not in the expected lexicon of the machine are typically substituted by some acoustically similar but nevertheless wrong words. The article discusses reasons for this undesirable behavior of the machine, describes some known examples of dealing with the unexpected words in human speech perception and their implications, and proposes an alternative architecture of ASR that could alleviate some of the problems with the unexpected acoustic inputs. Some published experimental results from using this alternative architecture are given.

**Keywords:** out-of-vocabulary words, automatic recognition of speech, parallel model of top-down and bottom-up human information extraction.

## 1 Introduction

There is enough evidence that biological systems pay more attention to unexpected events, which occurrence may represent a new opportunity or a new danger, than to the expected ones. A relatively well known is phenomenon of mismatch negativity that refers to a negative peak in brain event-related potential observed in human EEG signal about 150-200 ms after the onset of out-of-order stimulus in the train of repetitive auditory stimuli. The P300 positive peak in the EEG activity occurs about 300 ms after the presentation of unexpected stimulus in most sensory modalities – more unexpected the stimulus, larger is the peak. The N400 negative peak occurring about 400 ms after the presentation of unexpected stimulus that is hard to integrate semantically. It is most obvious in presentation of incongruous words in sentences and it is not limited only to the auditory modality (e.g. the picture of incongruous animal in the acoustically presented sentence may also trigger the N400). However, in this article, we will limit our discussion to unexpected sounds, namely to the unexpected words.

Sounds represent important interface with the outside world. In the form of speech, they provide for one of the most important cognitive functions, for the language communication. Without some common prior knowledge on both sides of the conversation, the human speech communication would not be possible. Thus, prior knowledge of a

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 1–15, 2011.

speaker and of a listener that are engaged in human speech communication plays a major role in getting the message through. However, any meaningful communication must also include at least some element of surprise. Without unexpectedness, there is no information conveyed. This characterizes the dilemma in use of prior information in speech communication. Skillful balancing of the predictable and the unpredictable elements in the message makes a master in communication. The same can be said about balancing these two elements in automatic recognition of speech (ASR). One of the most significant contributions of stochastic approaches to ASR is its systematic use of a language model, i.e. of a prior knowledge (context of the words). However, too much reliance on priors can be damaging. Incidents such as a cough causing the system to recognize the right command for a move by a machine in a chess-play (since that was the only reasonable move given the position in a game) [1] may still happen even with current state-of-the-art ASR systems. Just as in human speech communication, reaching the right balance between the reliance on priors and the reliance on the actual sensory input, is needed.

The problem comes up not only with out-of-speech sounds but also when dealing with words that are unknown to the recognition machine. These so called out-of-vocabulary words (OOVs) are source of serious errors in current large vocabulary continuous speech recognition systems (LVCSR). They are *unavoidable*, as human speech contains proper names, out-of-language and invented words, and also *damaging*, as it is known, that one OOV in input speech typically generates two or more word recognition errors [2].

Still, OOVs, since they are rare, usually do not have large impact on the word error rate (WER) of LVCSR. As such, they sometimes do not get the full attention in the current ASR culture where the WER is of the prime interest. However, the attention they rightly deserve. Rare and unexpected words tend to be information rich. Their reliable detection can significantly increase the practical utility of ASR technology in may important applications. It can indicate the need for an additional (machine or human) processing, and could allow for an automatic update of recognizer's vocabulary, or for description of the OOV region by phonemes.

## 2 Current ASR

ASR finds the unknown utterance $w$ by finding such a model $M(w)$ that with the highest probability $P(M(w))$ best accounts for the data $\mathbf{x}$, i.e. $w = \mathrm{argmax}(P(M(w_i)|\mathbf{x})$, where the search is carried out over all possible utterance models $w_i$ and is solved using the modified Bayes rule $w = \mathrm{argmax}(p(\mathbf{x}|M(w_i)^{\gamma}P(M(w_i)|\mathbf{x})^{1-\gamma})$, where likelihood $p(\mathbf{x}|(M(w_i))$ represents the conditional probability of the observed data, the prior knowledge is represented by the language model $P(M(w_i)$, and the modifications consist of ignoring the probability of the data $P(\mathbf{x})$, and of introducing the constant $\gamma$, which is used for *ad hoc* modifications of relative contributions of the prior knowledge, and that is set to optimize the performance on the training data.

In the large vocabulary conversational speech (LVCSR) the model of the utterance $M(w)$ typically represents a cascade of models of (conditionally independent) individual words, the likelihood of the whole utterance w is given by a product of likelihoods of all the individual words $M(w_i)$ that form the utterance model, i.e. $p(\mathbf{x}|M(w)) = \Pi p(\mathbf{x}|M(w_i)$. For the particular w to be chosen as the recognized one, both the likelihood $p(\mathbf{x}|M(w))$ and the language model $P(M(w))$ need to be reasonably high. Thus, low prior probability words are less likely to be recognized than the high probability ones. Even worse, when a particular word is not in the lexicon of the machine (so called out-of-vocabulary word (OOV)), its prior probability in by definition $P(M(w_{\text{OOV}}) = 0$. As long as the goal is to correctly recognize as many words as possible, i.e. to optimize the average word error rate, the use of the modified Bayes rule is fine. However, in communication of information by speech, not all words are created equal. The words that are highly predictable from the context of the discourse are typically less important than the words that are unexpected. So when the goal of the recognition is information extraction, it is important to correctly recognize high information value words. However, these are the words not favored by the language model since $p(M(w))$ for these words is low, and their low prior probabilities make the overall likelihood of the utterance that includes such a word also low. Further, the OOVs with their zero prior probabilities, are never recognized, and are substituted by at least one but more typically several acoustically similar shorter higher prior probability words. As already mentioned earlier, because of the interactions with the applied language model, the error in the OOV spreads to its neighborhood, and each OOV typically causes 2-3 word errors [2]. Further, unexpected words typically carry more information than the entirely predictable words, and substituting the OOV by other words might have disastrous consequences in information extraction applications.

This article argues that he way this prior knowledge is applied in ASR appears to be inconsistent with data on human language processing and suggest an alternative architecture of ASR that may help in alleviating this problem.

## 3 Unexpected Words in Human Communication

Two interesting lines of work, one in physiology and one in psychophysics of human speech communication, both related to human use of prior information in speech communication, have been discussed by J.B. Allen [3], and are reviewed below.

### 3.1 Physiological Evidence for the Parallel Combination of the Sensory and the Prior Knowledge (the Context) in Human Recognition of Speech

The N400, briefly mentioned in the Introduction, appears to be most closely related to high-level processes involved in processing of information in spoken language. The work of van Petten et al [4] deals with the issue of timing of the triggering of the N400 event by incongruous words that differ from the expected congruous word (e.g. "dollars" in the sentence "Pay with ....") either at their beginning (e.g. "scholars") or or at their ends (e.g."dolphins"). By their careful experimental design, where the instant of recognition of the individual words has been first established, they are able to demonstrate that the rhyming words ("scholars") trigger the N400 *earlier* than do the incongruous words with the correct first syllable ("dolphins"). This observation supports the notion of instantaneous integration of both the top-down prior context information and the bottom-up acoustic information.

However, the way the prior information is integrated in the current ASR is inconsistent with van Petten et al. data. In ASR, the prior $P(M(w))$ is invoked *globally* during the search for the best match of the whole utterance while the van Petten et al. data indicate that in human speech recognition, the prior knowledge is applied immediately (that is *locally*) at the instant of recognition of every word.

### 3.2 Psychophysical Evidence for the Parallel Prior Knowledge (Context) Channel in Human Recognition of Speech

Information about a word to be recognized is in bottom-up sensory data (acoustics) and on the top-down prior knowledge (context). As shown in [5], the error of recognition e in the context of other meaningful and related words, given by $e = e_a e_c$, where $e_a = (1 - p_a)$ represents the error of the acoustic channel and $e_c = (1 - p_c)$ represents the error of the hypothetical "context" channel that contributes to the correct word recognition with the probability $p_c$. Further, it is assumed that the $e_a$ and $e_c$ are proportional to each other, since the context words and the target word are being perceived under the same degradation (noise, etc.) and both the target word and the context words may be perceived in error. Then, the help the context words may provide in recognizing the message also depends on how well the context words themselves are recognized. Boothroyd suggests that the relation is $e_c = e_a^{k-1}$, where $k > 1$. This than implies $e = e_a^k$.

Boothroyd and Nittrauer [6] test their parallel combination model on perceptual data of Miller et al [7] who ran an experiment where they presented words from the closed set, first in the isolation and then forming the grammatically correct sentences, and evaluated accuracy of their recognition by human subjects in the increasing levels of noise. Accuracy of the recognition obviously decreased with the decreasing signal-to-noise ratio. As expected, the accuracy decreased slower for the words in the sentences than for the words presented in isolation. They show that data for errors in recognition of the out-of-context isolated words and of the in-context words in sentences, when plotted on the log-log scale, lay on two lines with the common origin of close to 100$\%$ error for the very noisy data but with different slopes. That means the in-context $e$ is related to the out-of-context $e_a$ though $e = e_a^k$ where $k = \log e_a / \log e > 1$. Boothroyd and Nittrauer [6] further also ran a series of their own experiments which also support the Boothroyd's model.

Later, Grant and Seitz [8] show that the k depends on the ability and the need of listeners to make use of the context. Thus, e.g. elderly listeners typically make better use of the context due to their better command of the language and also to compensate for their already partially impaired hearing. Further, even for the same listener, the k could be significantly higher in very noisy environments and much less when the speech is heard well and the context does not need to be heavily utilized. However, none of this changes the conclusion that in decoding the linguistic information in speech, the errors from the acoustic channel and from the context channel multiply. Since the errors always lay between 0 and 1, the final error $e$ is always less that either the $e_a$ or $e_c$.

Miller et al [7] and Boothroyd et al [6] support the existence of two statistically independent channels, one bottom-up acoustic channel and one top-down context channel. When an expected word comes and the acoustics is reliable, both channels indicate the same word. When the word is unexpected (e.g. out-of-vocabulary) or the acoustics is unreliable, the channels may disagree, each indicating different word. Still, the words will be recognized as long as the evidence in either the acoustic or the context channel is strong enough.

## 4 The Multiplication of Error in Parallel Processing Channels

The existence of parallel processing channels in decoding linguistic information in speech is not a minor issue. The parallel processing is the universal way of increasing reliability of any system. It allows for compensation of partial failures of some of the parallel channels by providing supporting evidence from the other (possibly more reliable) channels. Its existence in human processing of speech, as discussed later, suggest some major modifications of existing machine recognition architectures.

The observed multiplication of error rule is a consequence of a way the sensory input (acoustic speech signal) and the prior knowledge (context) are being used in decoding the linguistic message. This rule has been experimentally observed in early experiments in band-limited speech by Fletcher and his colleagues (see [3] for a review) and since it has been also observed for the acoustic and the context channel, supports its universal applicability in processing of sensory information. It implies that a reliable recognition (i.e. a small error) in any of the parallel channels yields a small final recognition error. That is the property which is highly desirable and that makes a lot of sense in information processing in biological systems. The following discussion, inspired and based loosely on the speculations of R. Galt on p. 148-151 of his notes [9].

When it comes to recognizing a word in an utterance, there are two ways of getting it right:

1. Guessing is from the context, using the knowledge of semantics of the situation as well as using the other already recognized words.
2. Getting it from the acoustic signal.

The words in the message can be divided into four categories:

1. There will be words that are supported by both the acoustics and the context. These words will be correctly recognized.
2. Some words will not be supported by the acoustics. These words will be correctly recognized because of their context
3. Some words that that are suggested by the acoustics will not be supported by the context. However, these will be correctly recognized because of the acoustic evidence.
4. Further, there will be words which are supported neither by the acoustics nor by the context and they will not be correctly recognized.

This, in the first three cases, the word is correctly recognized. It is only the last case that yields the error. Assuming that the message is very long, probabilities and rules for their combinations can be used. Then the speculation would be as follows:

Conditioned on the recognized word, the probability of correct recognition in the acoustic channel $p_a$ is conditionally independent on probability of the correct recognition in the context channel $p_c$. (Once the word is known, neither of these channels can provide any more information about the other channel). That allows for the use of the product of probabilities rule. These two ways of getting the information typically combine. The word can be recognized either from the acoustic channel or from the context channel or from both. These three situations are mutually exclusive. This allows for the use of the sum of probabilities rule. Thus, the compound probability of the correct recognition is given by a sum of different composite recognition probabilities under three mutually exclusive conditions:

1. When both the acoustics and the context are correct, the context channel is supporting the acoustic evidence. Under the assumption of a conditional independence of these two information channels, the probability of correct recognition is given by a product of probabilities $p_1 = p_a p_c$. This is happening when the acoustic signal is reasonably clean and well articulated and the speech segment being recognized follows well syntactic and semantic rules, i.e. the listener hears what she expects.
2. When the acoustic is in error but the context is correct, the probability of correct recognition is $p_2 = (1 - p_a)p_c$ .This happens when the acoustic segment is corrupted by some means (e.g. noise). In this case, as long as the context provides strong cues, the listener may rely on the context information and not on the unreliable acoustic evidence
3. When the acoustic is correct but the context is in error (e.g. unexpected word), the probability of correct recognition is $p_3 = p_a(1 - p_c)$. This happens when the acoustic segment does not fit the expectations of the listener. It may be because of mispronunciation or other error on the side of the talker but also it may be because the talker is trying to communicate something truly novel and therefore possibly also very important.

Since these three events are mutually exclusive, the probability of getting the word right is given by the sum $p = p_a p_c + p_a(1 - p_c) + (1 - p_a)p_c$ and the error of recognition is $e = 1 - p = 1 - p_a p_c - p_a(1 - p_c) - (1 - p_a)p_c = (1 - p_a)(1 - p_c) = e_a e_c$. This is of course the fourth case, when neither the acoustic nor the context support the correct recognition and the word is in error with the probability $p_4 = (1 - p_a)(1 - p_c)$, hence the product of errors rule!

This speculation tacitly assumes that the system knows when the correct recognition happens (no false alarms). These conditions could be possibly satisfied in many situations encountered in human information processing. However, it would be more difficult to assure that they are satisfied in machine processing as this would require reliable measure of confidence in the classification result.

## 5 The Proposal

Similarly as in human speech recognition, when ASR encounters an unknown word, this word may or may not be supported by the available prior knowledge (ASR language model and its lexicon). For identifying OOVs, it would be desirable to get an indication when the context does not support the acoustic evidence. Let's take an inspiration from the way human listeners may be processing the message as discussed above, and borrow the notion of the two information processing channels.

The information in the acoustic channel is $I_a$ and the information in the context channel is $I_c$. Combining the two channels, we get $I = I_a + I_c$. The information can be measured by entropy of the probability distribution of the estimates, i.e. $H = \sum \mathbf{p} \log_2 \mathbf{p}$. The information in both channels is $H = H_a + H_c$. We do not have a direct access to context channel. However, we have the final result of the combination of the information in both the context and the acoustic channels, measured by $H$. So, if we can estimate how the acoustic channel would do without the help of the sensory channel, i.e. to get $e_a$, then the information in the sensory channel could be inferred from the difference in the entropies $H$ and $H_a$, i.e. from the relative entropy $H_c = H - H_a$, as measured by the Kullback-Leibner divergence $D_{\mathrm{KL}}(\mathbf{p}||\mathbf{p}_a) = \sum \mathbf{p}_a \log_2(\mathbf{p}_a/\mathbf{p})$. To apply the scheme outlined above, we need $p$ and $p_a$. The $p$ represents the posterior probability of phonemes in the utterance estimated using the normal (strongly constrained) recognizer all its constraints on the language and the lexicon. As shown later, it can be derived from the result of the recognition. To obtain $p_a$, we also need to run another (weakly constrained) classifier that would not use the language and the lexical constrains. The item that is not supported by the prior knowledge would be indicated by a great discrepancy in posterior distributions from the strongly constrained and the weakly constrained classifiers.

### 5.1 How to Implement Our Scheme?

There is several of ways how to get the estimates of the posterior probabilities of elements of the sequential model M (most often phonemes or their parts).

Firstly, they can be obtained directly from the input data by a trained artificial neural net [10]. These estimates are constrained only by the constraints on its input (types of features and temporal span over which the features are derived) and by the training set for the ANN, such as the frequency of the occurrence (i.e. the prior probabilities) of the individual phonemes in the training set.

Secondly, they can be estimated from the results of the full recognition of whole strings of elements (i.e phonemes as they form words which in their turn form utterances). In that case, the estimates are strongly constrained by the language and lexical constraints. The posterior probabilities of phonemes can still be estimated after the recognition is done either by the Baum-Welch forward-backward estimation (e.g. [11] for the reference) or by counting the elements in the lattice of the recognized elements.

There are more possibilities in between these two extremes, which differ in the strength of constraints on the recognition. Thus, e.g. it is possible to design the recognizer which uses strings of phonemes instead of strings of words, thus eliminating the language constraints. The phoneme recognizer in its turn could still use the so-called phonotactics constraints (that represent the phoneme-level "language model") or could eliminate all context constraints and use only the within-phoneme constraints (i.e. the minimal length of the phoneme and the statistically identical distribution of features within the phoneme).

All these procedures could provide not only the most probable (i.e. the recognized) string of phonemes but also posterior probability vectors that represent probability distributions for all phonemes. These probability distributions can be used in deriving the KL divergence.

In the first case when the acoustic supports the context, the recognized word is identical to the word suggested by the acoustics, the $\mathbf{p}$ and the $\mathbf{p}_a$ will be similar and the KL divergence $D_{\mathrm{KL}}(\mathbf{p}||\mathbf{p}_a)$ will be small.

When the recognized word is different from the word suggested by the acoustics, the $\mathbf{p}$ and $\mathbf{p}_a$ will differ and the $D_{\mathrm{KL}}(\mathbf{p}||\mathbf{p}_a)$ will be large.

The second case would arise either when the context is overriding the acoustics (but the acoustics is reliable) or when the acoustics is unreliable and context is supplying the correct word. To differentiate between these two situations, we need the confidence on $\mathbf{p}_a$ and on $\mathbf{p}_a$.

Thus, to build the artificial system that would be capable of detecting OOV words, we would need:

1. Recognizing word from both the acoustics and the context,
2. recognizing the word only from the acoustics,
3. comparing the results from both recognition streams,
4. estimating reliability of recognition results in both parallel streams,
5. means for interpreting the indicated OOV it in terms of its parts (phonemes) that would allow for its description as well as for updating the lexicon.

The block diagram of the system is illustrated in Figure 1.

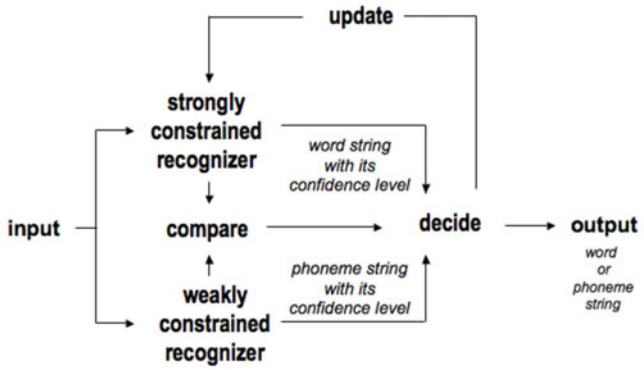

update
strongly constrained recognizer
word string with its confidence level
input
compare
decide
output
word or phoneme string
phoneme string with its confidence level
weakly constrained recognizer

**Fig. 1.** The block diagram of a system that could be used for detection and description of unexpected out-of-vocabulary words

The proposed system presents challenges in several aspects:

1. It calls for a reasonably reliable recognition of sub-word units (phonemes) from the acoustic signal alone, without the use of the prior information. Human capabilities are in this respect still far superior to any machine. While phoneme recognition

has been one of holly grails of speech research for decades, it has been largely abandoned (with a few notable exceptions) in favor of global stochastic matches for word sequences. To give a machine a chance, major improvements in acoustic processing are required.

2. It needs a way to compare results from both the strongly constrained and the weakly constrained recognition streams. Thus, the proper level of the comparison needs to be established, where the results of the recognition in both streams truly represent the estimated sequence of the underlying speech classes and are only minimally affected by other artifacts of the matching process. As it turns out, this is not a trivial problem.
3. It requires estimates of confidence in the recognition results in both the strongly constrained and the weakly constrained recognition streams. Even though the confidence measures are a topic of considerable research interest, reliable techniques are still lacking.

However, the topics are well defined, and addressing them is necessary in advancing the state-of-the-art in ASR.

## 6 Initial Results

Principles outlined above were first tested on the system shown schematically in Fig. 2. Posterior probabilities of phonemes for each individual speech frame (i.e. in equally spaced intervals of 10 ms) were derived from two levels of processing in a hybrid Hidden Markov Model (HMM) recognizer utilizing an artificial neural network (ANN)

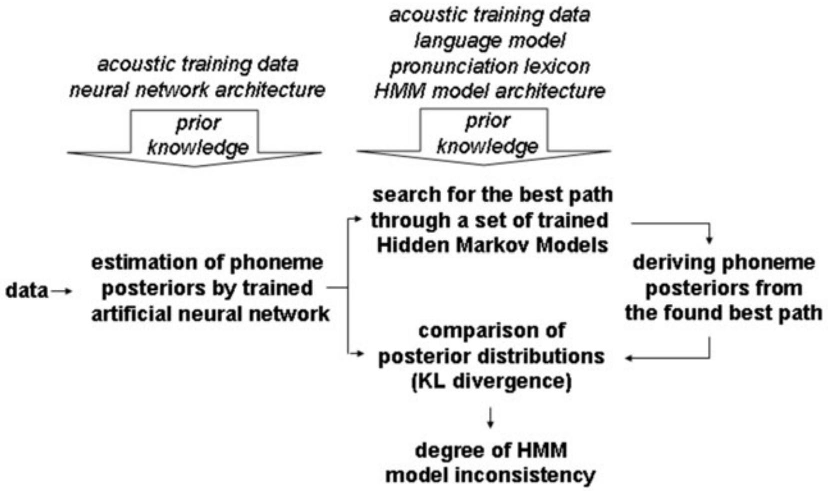

acoustic training data
neural network architecture
prior knowledge
acoustic training data
language model
pronunciation lexicon
HMM model architecture
prior knowledge
data
estimation of phoneme posteriors by trained artificial neural network
search for the best path through a set of trained Hidden Markov Models
deriving phoneme posteriors from the found best path
comparison of posterior distributions (KL divergence)
degree of HMM model inconsistency

**Fig. 2.** Discovery of out-of-vocabulary words using hybrid HMM-NN ASR system, in which the out-of-context posterior probabilities estimated by the ANN are also directly used in the constrained search for the best model sequence

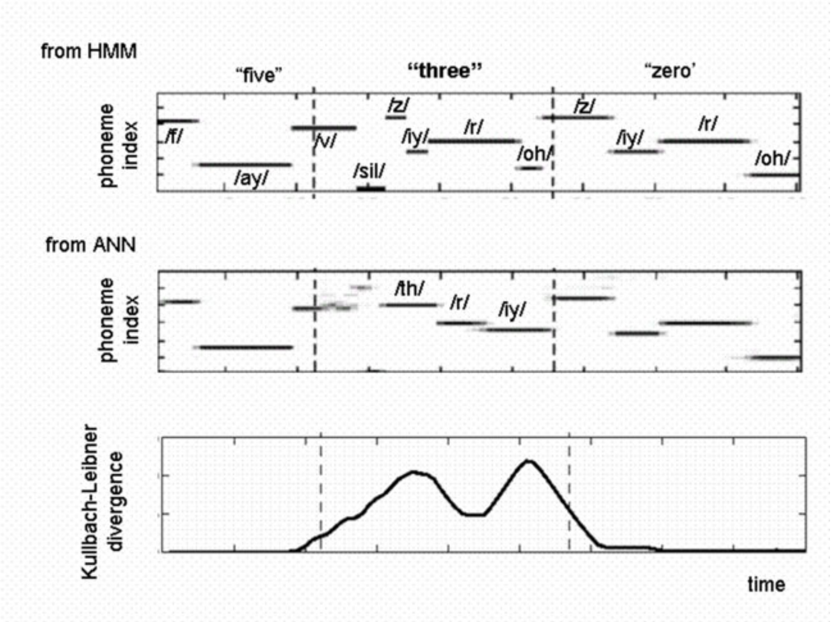


**Fig. 3.** Posterior probabilities (posteriograms) of phonemes estimated by an HMM-based system (the upper part of the Figure), and by ANN (the middle part of the Figure). In this example, the HMM model inconsistency was introduced by removing the word `three' from the recognizer vocabulary. The correct phoneme sequence for the word `three' is misrepresented in the HMM-derived posteriogram (replaced by a sequence /z/iy//r//oh/ of the in-vocabulary word `zero'). The ANN derived probabilities indicate in this case the correct sequence /th//r//iy/ for the out-of-vocabulary word `three'. Comparison of the respective posterior probability density functions by evaluating their relative entropy (KL divergence), shown smoothed by 100 ms square time window as a function of time in the lower part of the figure, indicates HMM model inconsistency in the neighborhood of the out-of-vocabulary word `there' (The figure is adopted from [12] that also gives more details).

probability estimation (HMM/ANN ASR) [2]. The weakly-constrained phoneme probabilities are estimated by the trained ANN. These are subsequently being used in the search for the most likely stochastic model of the input utterance. Thus, one set of posterior probabilities was obtained directly from the ANN (weakly-constrained classifier), another set comes from the Baum-Welch estimation procedure that provides phoneme posteriors derived with the use of the prior constraints, in this case from knowledge of the expected lexicon (strongly constrained classifier). The comparison was done by evaluation measuring KL divergence between the posterior probability distributions in the sensory and context channels.

An example of in context and out of context posteriors, and the smoothed divergence as a function of time is shown in Fig. 3. The utterance contains 'five three zero' where the word 'three' represents an unexpected word, not present in the vocabulary. The upper part shows the out-of-context posteriors, the middle part the in-context posteriors, and the lower part shows the smoothed KL divergence between two. As it can be seen, there is a region with major divergence corresponding to the word ‘three’ (which is marked roughly by dashed lines). An example of a typical result is shown in Fig. 3. As seen in the lower part of the Fig. 3, an inconsistency between these two information streams could indicate unexpected out-of-vocabulary word.

11 digit OGI digits database was used for the evaluation of the technique. Each of the the words was introduced individually one after another as an unexpected word by removing it from the vocabulary and compared our technique with a group of state-of-the-art conventional posterior based confidence measures. The new approach yielded noticeably larger area under the ROC curve (much better trade off between true and false alarms). More detail can be found in [12].

## 7 Next Steps

The proposed technique has been also investigated on the detection of OOV segments in the output of large vocabulary continuous speech recognition (LVCSR) system on Wall Street Journal (WSJ) at the 2007 Summer Research Workshop at the Johns Hopkins University. All aspects of the system have been studied, with most effort devoted to 1) recognizing speech with a minimal use of the context, 2) comparing estimates from the “strongly-constrained” (i.e. both the acoustics ad the context constrained) and the “weakly-constrained” (mostly acoustics) recognition streams 3) the confidence measures, 4) phoneme recognition without use of any context. The data material consisted of WSJ data, down-sampled to 8 kHz, with about 20% of least frequent words left out from the lexicon, thus emulating the targeted OOVs. In addition to the test data-set, a development set (used for a training of some of data-guided techniques) has been also created. A state-of-the-art LVCSR recognizer has been used as the strongly-constrained recognizer. The weakly-constrained recognizer was the same LVCSR system modified for recognition of phonemes (rather than words). Both recognizers were trained on the independent telephone-quality data and not on the targeted WSJ data. Additionally, the strong constrains were also induced on the recognized phoneme string by a transducer-based system from Microsoft Research. A number of comparison techniques have been investigated in addition to many state-of-the-art confidence measures derived from both the strongly-constrained and the weakly-constrained recognition streams. In the final system, results from most of the investigated techniques have been fused using a state-of-the-art classifier from Microsoft Research. The progress has been evaluated by comparing the developed error-detection techniques to the Cmax technique [13]. Results of the comparison support the effectiveness of the technique. More details of this effort can be found in [14], [15].

## 8 Summary of More Recent Results

The new system replaced KL-divergence by a trained neural network (NN) based classifier. This NN based OOV word detection was applied to noisy, lower quality telephone speech (CallHome, Eval01, Fisher) to show the robustness of the approach. Further, the two-way (in-vocabulart/out-of-vocabulary) classification was replaced by a four-way

NN classifier (IV correct, IV incorrect, OOV, silence) [16]. Finally, the research focused on the detection of most important frequently re-occurring unexpected words. To achieve this, we ran our OOV detection system on telephone calls and lectures centered around a certain topic [17].

Final goal is to obtain descriptions of OOV words - putting special emphasis on repeatedly occurring OOVs. The task is now to detect the time span of the reference OOV word as completely and precisely as possible. Since the NN-based OOV detection system offers no description/boundaries of the OOV, a integrated a hybrid word/sub-word recognizer [19] was integrated into the system. Comparing to a conventional LVCSR, such system does not substitute an OOV by best-matching IV; it can fall back to its sub-word model and thus retrieve a description of the word in terms of sub-word units. Here the boundaries for OOV words were estimated more accurately than with the NN-based system.

Experiments were also ran with much larger decoding vocabulary and a data set consisting of a collection of topic-specific TED talks, in which we find a reasonably high number of information-rich OOV words [19]. It represents a more realistic scenario, since only naturally appearing and harder to detect OOVs are targeted.

Another extension of the proposed approach was with follow-up actions that can be taken after the detection of an OOV to analyze the newly discovered words and to recover from the mis-recognitions. The goal is to avoid mis-recognitions in the presence of rare words by designing a system that is open-vocabulary and that can learn with its usage. The hybrid word/sub-word recognizer solves the OOV localization and obtains its phonetic description – the detected phoneme sequence in the detected time span - in an integrated way [20]. Given the location and phonetic description of an OOV, one possible action is to recover the orthographic spelling of the OOV. OOV spelling recovery [19] can successfully recover many OOVs, lowering the word error rate and reducing the number of false OOV detections.

Aiming at topic-specific repeating OOVs, a similarity scoring and clustering of detected OOVs was introduced. A similarity measure [17] based on aligning the detected sub-word sequences were developed, which serves to identify similar candidates among all OOV detections. A new form of word alignment is introduced, based on aligning the OOV to sequences of IVs/other OOVs, which retrieves a higher-level description of the OOV, in the sense of word relations. (e.g. being a compounded word or a derivation of a known word).

### 8.1 Some Thoughts for the Future

Machine learning (ML) dominates current approaches to automatic recognition of speech (ASR). Training data (both acoustic speech data and text) are used to build a model of a spoken language. In a recognition stage, incoming speech is compared to the model and the best matching elements of the model determine what has been said. Many innovative ML techniques have been developed and new ones are appearing in a rapid pace. No matter how sophisticated, all ML techniques have one thing in common – they require training data. Common belief in ASR community is that most significant improvements in performance of ASR system come from more data. Implicit is a tacit assumption that speech to be recognized comes from the same distribution as the data on which the machine was trained.

Problems occur when this assumption is violated. That happens more often than not. There are “unknown unknowns” not only in military situations but also in speech. In this work we started to address words that are not in a lexicon of a machine or occur with low probability. However, there are also unexpected distortions of a signal and noises, unknown accents and other speech peculiarities, that are sources of an unexpected variability that create problems for the current ASR. The problem is inherent to machine learning and will not go away unless alternatives to extensive reliance on false beliefs of unchanging world are found. It appears that some fundamental flaw in machine design needs to be corrected to succeed in dealing with unexpected sources of variability in the information carrying signal.

Some suggestions for the system of the future may come from studies of human hearing system. In the work presented in this article we studied two information streams, on with stronger prior constraints and one with weaker ones. Human auditory cortex contains several millions of cortical neurons. Firing cortical firing rates are somewhere of the order of tenths per second. However, at the first stages of the processing on the level of the auditory nerve, there are about two orders of magnitude fewer neurons but the firing rates are two orders of magnitude faster [21]. Fewer neural connections between the hearing periphery and the cortex are going from the periphery to the cortex, many more are going from the cortex down to the periphery. At its origin, the stimulus serves rich but relatively narrow information stream. As the stimulus goes through processing stages of a perceptual system, the stream of information gets progressively sparser in time but richer in features. The processing relies heavily on feedback strategies, where higher processing stages influence processing in stages below.

Why is the system configured in this way? Let’s speculate. At the end, each cortical neuron provides for a separate channel in processing an incoming auditory stimulus. Each channel serves the information about one auditory item each 100-200 ms. It is well accepted that mammalian (i.e. also human) hearing system provides for separation of signal components with different carrier frequencies, different rates of change and different levels of spectral detail [22]. Thus, each item is described in millions ways, using information from different parts of the auditory spectrum, with different spectral and temporal detail, and with different degrees of prior expectations. The final decision about the item then could be made by comparing the individual processing streams and selectively using the most reliable information from the individual streams, possibly entirely alleviating the unreliable streams from the final decision [23], [24], [25].

**Acknowledgement.** This work was partly supported by European project DIRAC (FP6-027787) and by Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20015, where the concept of parallel processing streams is being developed further to combat unexpected signal distortions [23], [24], [25]. Contributions of members of the DIRAC research team to the research presented in this paper are acknowledged.

## References

1. Klatt, D.H.: Review of the ARPA speech understanding project. J. Acoust. Soc. Am. 62, 1345–1366 (1977)
2. Chase, L.L.: Error-Responsive Feedback Mechanism for Speech Recognizers, PhD Thesis, Carnegie-Mellon University
3. Allen, J.B.: Articulation and Intelligibility. Morgan & Claypool (2005)
4. Van Petten, C., et al.: Time course of word identification and semantic integration in spoken language. J. Experimental Psychology: Learning, Memory, and Cognition 25(2) (1999)
5. Boothroyd, A.: Speech perception and sensorineural hearing loss. In: Ross, M., Giolas, G. (eds.) Auditory Management of Hearing-Impaired Children, University Park, Baltimore, MD (1978)
6. Boothroyd, A., Nittrouer, S.: Mathematical treatment of context effects in phoneme and word recognition. J. Acoust. Soc. Am. 84(1), 101–114 (1988)
7. Miller, G.A., Heise, G.A., Lichten, W.: The intelligibility of speech as a function of the context of the test material. J. Exp. Psychol. 41, 329–335 (1951)
8. Grant, K.W., Seitz, P.F.: The recognition of isolated words ands words in sentences: Individual variability in the use of sentence context. J. Acoust. Soc. Am. 107(2)
9. Rankovic, C., Allen, J.B.: Study of Speech and Hearing in Bell Telephone Laboratories: The Fletcher Years. In: CD ROM with Correspondence, Internal Reports and Notebooks of R. Galt (1917-1933). Acoustical Society of America, Melville (2000)
10. Bourlard, H., Wellekens, C.J.: Links between Markov Models and Multilayer Perceptrons. In: Touretzky, D. (ed.) IEEE Conference on Neural Information Processing Systems, 1988, Denver, CO, pp. 502–510. Morgan-Kaufmann Publishers, San Francisco (1989)
11. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press, Cambridge (1998)
12. Ketabdar, H., Hannemann, M., Hermansky, H.: Detection of Out-of-Vocabulary Words in Posterior Based ASR. In: Proceedings of the International Conference on Spoken Language Processing, Antwerp, Belgium (2007)
13. Wessel, F., et al.: Confidence measures for large vocabulary continuous speech recognition. IEEE Trans. Speech and Audio Processing 9(3), 288–298 (2001)
14. White, C., et al.: Confidence Estimation, OOV Detection And Language ID Using Phone-To-Word Transduction And Phone-Level Alignments. In: Proc. ICASSP (2008)
15. Burget, L., et al.: Combination Of Strongly And Weakly Constrained Recognizers For Reliable Detection Of OOVs. In: Proc. ICASSP (2008)
16. Kombrink, S., et al.: Posterior-based Out of Vocabulary Word Detection in Telephone Speech. In: Proc. Interspeech 2009, Brighton, U.K (2009)
17. Hannemann, M., et al.: Similarity scoring for recognized repeated Out-of-Vocabulary words. In: Proc. Interspeech 2010, Makuhari, Japan (2010)
18. Szöke, I., Fapso, M., Burget, L., Cernocky, J.: Hybrid Word-Subword Decoding for Spoken Term Detection. In: SSCS 2008 - Speech Search Workshop at SIGIR (2008)
19. Kombrink, S., et al.: Recovery of rare words in lecture speech. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS (LNAI), vol. 6231, pp. 330–337. Springer, Heidelberg (2010)
20. Kombrink, S.: OOV detection and beyond. In: DIRAC workshop at ECML/PKDD, Barcelona (2010)
21. Tobias, J.V.: Foundations of Modern Auditory Theory. Academic Press, London (1970)
22. Mesgarani, N., et al.: Phoneme representation and classification in primary auditory cortex. Acoust. Soc. Am. 123, 899–909 (2008)
23. Mesgarani, N., et al.: Toward optimizing stream fusion in multistream recognition of speech. J. Acoust. Soc. Am. 130(1), EL14–EL18 (2011); (5 pages)

24. Hermansky, H., et al.: Performance Monitoring For Robustness In Automatic Recognition Of Speech. In: Proc. Symposium on Machine Learning in Speech and Language Processing, Bellevue, Washington, USA (June 2011)
25. Mesgarani, N., Thomas, S., Hermansky, H.: Adaptive Stream Fusion in Multistream Recognition of Speech. In: Proc. Interspeech (2011)

# A Cloud on the Horizon

Mark Epstein[1,2]

[1] IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598, U.S.A.
meps@us.ibm.com
[2] Google Inc.
1600 Amphitheatre Parkway, Mountain View, CA, U.S.A

**Abstract.** In this talk, I will present an overview of many areas in speech and language processing that Google is researching, developing and deploying to users. As the amount of information on the web grows, difficult problems in speech, text, information retrieval and natural language processing are now being solved and new challenges are appearing. For example, Google now builds speech recognition language models for a one million word vocabulary from billions of n-grams. By using data-driven approaches, Google is improving it's ability to deliver information to global users of computers, tablets and smartphones.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, p. 16, 2011.

# A Novel Lecture Browsing System Using Ranked Key Phrases and StreamGraphs

Martin Gropp, Elmar Nöth, and Korbinian Riedhammer

Lehrstuhl für Informatik 5 (Mustererkennung)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstr. 3, 91058 Erlangen, Germany
korbinian.riedhammer@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de

**Abstract.** A growing number of universities offer recordings of lectures, seminars and talks in an online e-learning portal. However, the user is often not interested in the entire recording, but is looking for parts covering a certain topic. Usually, the user has to either watch the whole video or "zap" through the lecture and risk missing important details. We present an integrated web-based platform to help users find relevant sections within recorded lecture videos by providing them with a ranked list of key phrases. For a user-defined subset of these, a StreamGraph visualizes when important key phrases occur and how prominent they are at the given time. To come up with the best key phrase rankings, we evaluate three different key phrase ranking methods using lectures of different topics by comparing automatic with human rankings, and show that human and automatic rankings yield similar scores using Normalized Discounted Cumulative Gain (NDCG).

**Keywords:** key phrases, ranking, visualization, browsing, e-learning.

## 1 Introduction

A growing number of universities offer e-learning material to both their students and, to some extent, the public. Aside from lecture slides or work sheets, many schools provide audio or video recordings of lectures, seminars and talks. Software solutions like *iTunes U*[1] or *OpenCast*[2] help with the recording, storage and organization of the data.

Most e-learning sites provide the user only with a catalog of audio and video recordings, sometimes annotated with short descriptions, tags or user comments. The documents themselves are presented as is, usually an audio or video file with play, pause, rewind and scroll controls which is sufficient for a student who is interested in the whole recording.

However, the same archives are often used as supplemental material when preparing a class project or studying for an exam. In these cases, the user is interested in whether or not a certain topic or key phrase is mentioned in the recording and if so, when and

[1] http://www.apple.com/education/itunes-u

[2] http://www.opencastproject.org

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 17–24, 2011.

in what context it occures. As an example, consider a student refreshing a class on machine learning who is interested in regression and classification. Without the information mentioned above, he or she would either have to listen to all recordings, which is very time consuming, or trying to "zap" through all the recordings to spot some key words, hoping to catch all relevant parts of the lecture.

In this work, we first evaluate three different key phrase ranking strategies in a small user study that was part of a student thesis [3]. In a next step, we introduce an integrated interactive web-based platform which provides the user with the recording, a ranked list of important key phrases and, for a subset, a visualization of when these key phrases appear, which can be used to navigate within the recording. Together with the possibility to manually add, delete or re-rank key phrases, this integrated tool can greatly increase the use of the recordings and contributes to making e-learning easier and more efficient.

## 2 Related Work and Motivation

The motivation for this work is to step away from extractive summarization of spoken language as it is limited in terms of readability and involves the risk of omitting important details. Instead, we provide the user with a tool to find all the information he or she needs within a short period of time. The user is presented with the original data to avoid recognition errors and confusions due to utterances extracted without context.

The objective of the proposed integrated platform to is to help the user find the information he or she is looking for in the lecture. This task is closely related to (query based) extractive summarization which is the concatenation of salient utterances. In [11], a tool for interactive meeting summarization was presented. The user is provided with an initial set of weighted key phrases which are used to compute an extractive summary. The user can then modify the key phrases and their weights to produce summaries that provide the requested information. Though the interface was never thoroughly evaluated, follow up work [12] confirmed that well weighted key phrases can be used to compute very good extractive summaries compared to human abstracts.

Similarly, we extract and rank key phrases to provide an initial overview of the lecture but choose a graphical representation (see Sec. 6) instead of an extractive summary. Key phrase extraction is traditionally divided in supervised (i.e., a previously trained classifier decides whether or not a phrase is salient, e.g. [6]) and unsupervised methods which do not require prior training. Recent works on meeting summarization utilize part-of-speech n-grams, lexical chains, or graph-based methods, e.g. [12,7]). More recent works suggest to extract key phrases using a ranking approach [5] utilizing Learning-to-Rank methods as found in the information retrieval community [8].

## 3 Data

The BASE Corpus[3] consists of 160 lectures and 40 seminars recorded in a variety of departments (video-recorded at the University of Warwick and audio-recorded at the University of Reading). It contains 1,644,942 tokens in total (lectures and seminars).

[3] http://www2.warwick.ac.uk/fac/soc/al/research/collect/base

Holdings are distributed across four broad disciplinary groups, each represented by 40 lectures and 10 seminars. In this work, we focus on the "Arts and Humanities" lecture series *ahlct*, namely lecture *008* on Huckleberry Finn, and lecture *009* from the introductory Assembler lecture series *pslct*.

# 4 Key Phrases

## 4.1 Candidate Extraction

Fig. 1 shows an overview of the key phrase extraction process. The input data, either the output of a speech recognition system (ASR) or, as for the BASE corpus, a manual transcription, is split in chunks by the sentence detector using annotations like punctiations or pauses (transcription) or prosodic cues (ASR). The Tokenizer prepares the input for the Part-of-Speech (PoS) tagger [9]. Word form normalization [10] and a stop words filter (about 900 words containing conversational speech artifacts) finalize the pre-processing.

The candidate selection is taken from [12] and can be summarized as matching PoS patterns against a regular expression allowing certain sequences of tags modeling noun phrases. Example key phrase candidates extracted from lecture *ahlct008* are *"Huckleberry Finn", "Mark Twain", "Tom Sawyer", "American literature"*, and *"civil war"*.

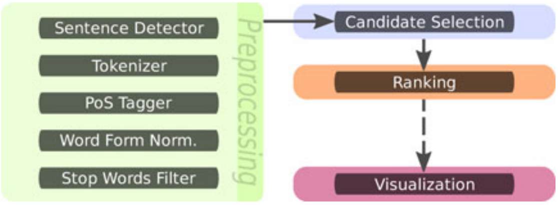


**Fig. 1.** Key phrase extraction process

## 4.2 Ranking

**No Language Model.** Similar to [12], we design a heuristic ranking function which combines the frequency ($n$) of a candidate phrase with its n-gram length using a weighting function $w$ that emphasizes phrases of length 2 or 3

$$\text{f x len}(g) = n \cdot w(n_t) \quad , \quad w(x) = x \cdot e^{-\frac{1}{5}x^{3/2}} \tag{1}$$

where $n_t$ is the number of words within the phrase with POS tags indicating nouns, foreign words, numbers, adjectives or gerunds.

**Corpus Specific Language Model.** A common feature in information retrieval is the term frequency (TF) multiplied by an inverse document frequency (IDF) giving a notion of how document-specific a word or phrase is. The IDF values are estimated on a representative document collection of the target domain. Here, we consider an ideal setup where we estimate the IDF values on all lectures of a series, i.e., we get corpus specific

IDF values for the series *ahlct* and *pslct* which can be integrated in the frequency based ranking as

$$\text{tfidf x len}(g) = n \cdot \text{IDF}(g) \cdot w(n_t) \tag{2}$$

**General Background Language Model.** A more general approach is to compare phrase occurrences to a general background language model. We estimate phrase distribution probabilities on the *British National Corpus* [1] and compare them using a point-wise Kullback-Leibler (KL) divergence [13]

$$\text{KL}(g) = p(g) \cdot \log_2 \frac{p(g)}{q(g)} \tag{3}$$

where $p(\cdot)$ and $q(\cdot)$ are the document and background phrase probabilities. Note that there is no heuristic correction for phrase length for the same reason as in KS.

# 5 Evaluation

## 5.1 Setup

For the evaluation of the system human raters were given transcripts of the lectures. Unlike the direct output of speech recognizers, these transcripts were first checked for recognition errors, divided into sentences and meaningful paragraphs, and had punctuation added. It can be assumed that these superficial changes, while dramatically improving readability, do not affect the way humans understand a text, and therefore have no impact on the key phrases identified by the rater.

The raters were asked to read the lecture and produce a sorted list of 20 key phrases they thought were most suitable for representing its content. The relevance of these phrases should be rated on a scale ranging from 1 (very relevant) to 6 (extraneous)[4]. Any unrated phrase was assumed to belong to the worst category and given a 6.

An analysis of the phrases selected by our algorithm showed that there were almost no good phrases that were not ranked among the best 15 by at least one of the methods. So, in order to make things easier for the raters, they were provided with a list of candidates to choose from which consisted of the best 20 phrases according to each of the employed relevance measures. This typically resulted in about 50 alphabetically sorted items. Under the assumption that all sensible candidates appear on this list, this method should have no effect on the results. However, the raters were encouraged to add more phrases from the text.

## 5.2 Evaluation Measure

The Normalized Discounted Cumulative Gain (NDCG) [4] rewards placing “valuable” items at the top of a retrieved list. Every phrase $\varphi_i$ is assigned a “gain” (the more relevant, the higher) multiplied with a discount factor based on its position in the list (the further to the back, the lower).

[4] These correspond to the German school grading system which has turned out to produce more homogeneous results than other scales.

In order to emphasize the top ranks, we use an exponential function to map the grade assigned by the rater to a gain value:

$$\text{gain}(\varphi_i) = 2^{(6-\text{grade}_i)/5} - 1 \tag{4}$$

The discount function suggested by J"arvelin and Kek"al"ainen is the reciprocal logarithm, but since there is no particular reason for this exact function, we use $1/\log_2(1+\text{pos})$ to avoid any special treatment for the first item. The base of the logarithm can be varied depending on how much the top ranks should be emphasized.

The DCG is then simply the sum over the discounted gains of all phrases

$$\text{DCG}_N = \sum_{i=1}^{N} \frac{\text{gain}(\varphi_i)}{\log_2(1+i)} \quad , \quad \text{NDCG}_N = \frac{\text{DCG}_N}{\text{ideal DCG}_N} \tag{5}$$

which is then normalized by division by the ideal DCG (of a sorted list).

### 5.3 Results

The lectures *ahlct008* and *pslct009* were each evaluated by five human raters (computer science students). We are now interested in the quality of both human and automatic rankings.

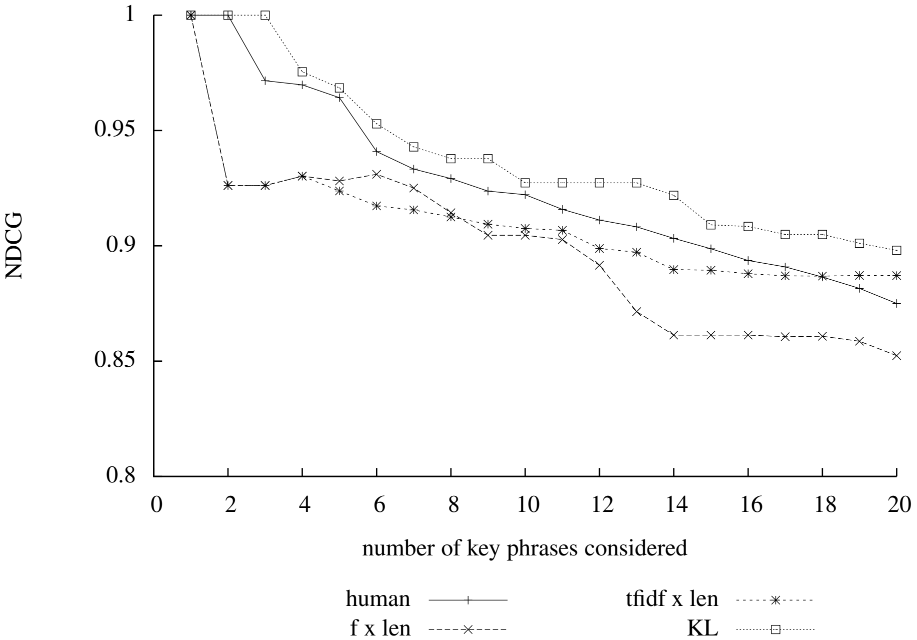


**Fig. 2.** NDCG scores for lecture *ahlct008*

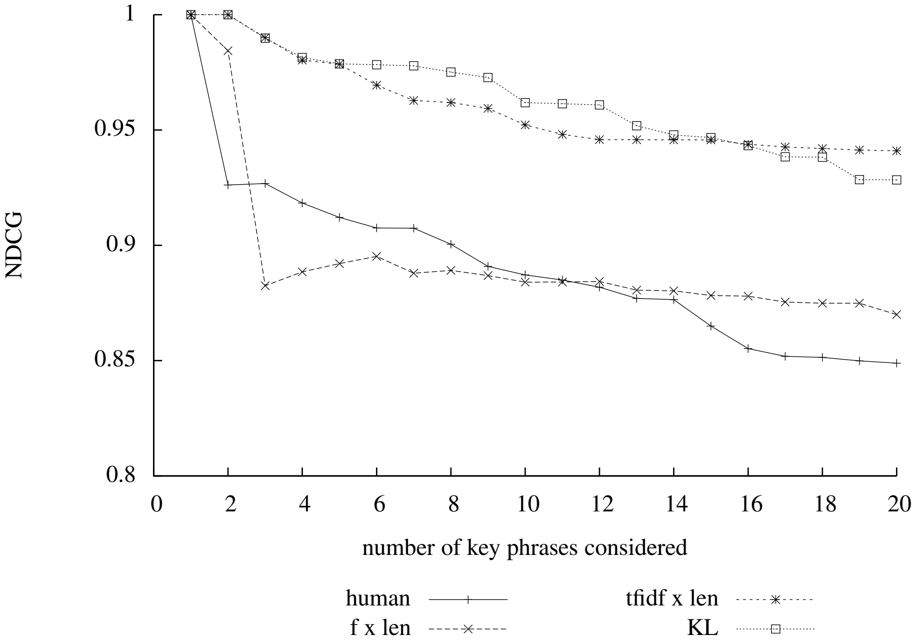


**Fig. 3.** NDCG scores for lecture *pslct009*

To ensure a fair evaluation, we select one human rater's relevance scores to calculate the NDCG scores for the remaining human and the automatic rankings. This is repeated five times to use each rater's relevance scoring once. Finally, all human NDCG scores are averaged to single NDCG score (human). Similarly, a mean is computed for each automatic ranking method (f x len, tfidf x len, KS and KL).

Fig. 2 and 3 show the NDCG scores for the lectures *ahlct008* and *pslct009*. The Y axis represents the evaluation measure, where 1 is the best achievable value. The X axis specifies the number of key phrases considered for the evaluation (beginning with the top ranked key phrase).

As expected, the human rankings (continuous line) receive consistently good scores which decrease for the number of key phrases considered. This makes sense as the raters strongly agree on the really important key phrases but not necessarily agree on the less important ones.

For the completely unsupervised rankings, the KS ranking could not produce satisfactory rankings. The original idea that uniform phrase distribution corresponds to non-salience may be flawed as lectures tend to have a certain topic, thus important phrases may very well be uniformly distributed. The strongly different behavior of the ranking for the two different lectures however suggests that this the phrase distribution may be both topic and lecture style dependent.

The general observation is that, considering a useful number of key phrases per lecture, e.g. 5 to 10, the automatic rankings are comparable to human rankings, i.e., it is hard to tell the difference between an automatic and a human ranking. Furthermore, integrating suitable language model information helps to keep the automatic ranking consistent with the human rankings. For the humanities lecture *ahlct008*, the general background language model seems to be more adequate, while the technical *pslct009* can benefit from the corpus specific information.

## 6 Integrated Browsing System

Though the key phrase extraction and ranking produce reliable results, just a textual representation is not sufficient to get an overview of the whole lecture. Thus, we integrate them into a browsing interface as depicted in Fig. 4: The StreamGraph [2] on the bottom left visualizes when important key phrases occur and how prominent they are at the given time by mapping a key phrase to one colored wave. This can be used to navigate within the video: by clicking on the desired position on the StreamGraph (horizontal for time, vertical for phrase and dominance), the video play back begins a few seconds before the occurrence of the requested phrase. The list on the right shows the available key phrases and controls which phrases should be included in the graph, usually about five. Furthermore, the user can remove existing or add further key phrases as desired.

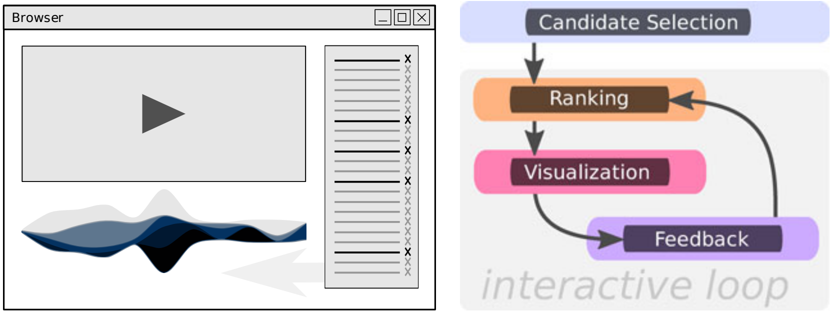


**Fig. 4.** A mock-up of the integrated browsing system. The video can be controlled by clicking into StreamGraph below. The list on the right shows the available and key phrases (displayed phrases in bold face).

Once the system is in regular use, statistics about favored, deleted and added key phrases can be collected which are the basis for combining existing ranking methods by Learning to Rank.

## 7 Summary

In this work, we compared four unsupervised methods to rank automatically extracted key phrases. We conducted a small user study on lectures of different topics and could show that the best automatically ranked key phrases are of similar quality as human rankings, especially for shorter lists of key phrases.

Furthermore, we motivated and described a web-based platform for lecture video browsing. In addition to the video, we provide an automatically extracted and ranked list of key phrases. A user defined selection is displayed by a StreamGraph that visualizes when the phrases occur and how prominent they are at the given time. This allows the user to quickly find the information he or she needs and provides a more natural interface in contrast to extractive summarization.

The interface allows to collect statistics about certain user interactions, e.g. which key phrases are visualized most in the StreamGraph or which key phrases were added or deleted. These data can then help to develop better unsupervised methods or build the basis for supervised Learning-to-Rank methods.

**Acknowledgments.** Part of the transcriptions used in this study come from the British Academic Spoken English (BASE) corpus project. The corpus was developed at the Universities of Warwick and Reading under the directorship of Hilary Nesi and Paul Thompson. Corpus development was assisted by funding from BALEAP, EURALEX, the British Academy and the Arts and Humanities Research Council. Part of this work was supported by the European regional development fund (ERDF) under STMWVT grant IUK-0906-0002 in cooperation with the Medav GmbH.

## References

1. Burnard, L. (ed.): Reference Guide for the British National Corpus. Research Technologies Service at Oxford University Computing Services (February 2007)
2. Byron, L., Wattenberg, M.: Stacked Graphs – Geometry & Aesthetics. IEEE Transactions on Visualization and Computer Graphics (TVCG) 14(6), 1245–1252 (2008)
3. Gropp, M.: Key Phrases for the Textual and Visual Summarization of Academic Spoken Language. Studienarbeit Informatik, Dept. Informatik 5, Univ. Erlangen-Nuremberg (2010)
4. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20(4), 422–446 (2002)
5. Jiang, X., Hu, Y., Li, H.: A ranking approach to keyphrase extraction. In: Proc. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 756–757 (2009)
6. Liu, F., Liu, F., Liu, Y.: Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion. In: Proc. IEEE Workshop on Spoken Language Technologies (SLT), pp. 181–184 (2008)
7. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Proc. NAACL-HLT, pp. 620–628. ACL, Stroudsburg (2009)
8. Liu, T.Y.: Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval 3 (2009)
9. Phan, X.: CRFTagger: CRF English POS Tagger (2006), http://crftagger.sourceforge.net
10. Porter, M.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
11. Riedhammer, K., Favre, B., Hakkani-Tür, D.: A Keyphrase Based Approach to Interactive Meeting Summarization. In: Proc. IEEE Workshop on Spoken Language Technologies (SLT), pp. 153–156 (2008)
12. Riedhammer, K., Favre, B., Hakkani-Tür, D.: Long Story Short – Global Unsupervised Models for Keyphrase Based Meeting Summarization. Speech Communication 52(10), 801–815 (2010)
13. Tomokiyo, T., Hurst, M.: A Language Model Approach to Keyphrase Extraction. In: Proc. ACL Workshop on Multiword Expressions, pp. 33–40 (2003)

# Addressing Multimodality in Overt Aggression Detection

Iulia Lefter[1,2,3], Leon J. M. Rothkrantz[1,2], Gertjan Burghouts[3], Zhenke Yang[2], and Pascal Wiggers[1]

[1] Delft University of Technology, The Netherlands
[2] The Netherlands Defense Academy
[3] TNO, The Netherlands

**Abstract.** Automatic detection of aggressive situations has a high societal and scientific relevance. It has been argued that using data from multimodal sensors as for example video and sound as opposed to unimodal is bound to increase the accuracy of detections. We approach the problem of multimodal aggression detection from the viewpoint of a human observer and try to reproduce his predictions automatically. Typically, a single ground truth for all available modalities is used when training recognizers. We explore the benefits of adding an extra level of annotations, namely audio-only and video-only. We analyze these annotations and compare them to the multimodal case in order to have more insight into how humans reason using multimodal data. We train classifiers and compare the results when using unimodal and multimodal labels as ground truth. Both in the case of audio and video recognizer the performance increases when using the unimodal labels.

## 1 Introduction

It comes very easy for people witnessing a scene to judge whether it is aggressive or not, and given the context, whether such behavior is normal or not. In their judgment they rely on any clue that they might get from sound, video, semantics and context. The exact process by which humans combine these information to interpret the level of aggression in a scene is not known. When one modality is missing, the decision becomes harder. With our research we are trying to mimic human decision process by means of computer algorithms that are able to discriminate between different patterns.

The problems with automatically processing multimodal data start already from the annotation level. Should we annotate multimodal data in a multimodal way? Do we need extra unimodal annotation? Typically unimodal processing techniques are used for training models based on multimodal labeling. Yet in many cases an event is apparent only in one of the modalities.

In this paper we take an approach resembling the human model of perception of aggressive scenes as depicted in Figure 1. Historically, human perception has been viewed as a modular function, with the different sensor modalities operating independently of each other and then apply semantic fusion. To emulate human processing we researched unimodal scoring systems. Therefore we use both unimodal and multimodal annotations. That is to say, we requested human annotators to score the data using only

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 25–32, 2011.

video recordings, only sound recordings or using both. These unimodal annotations give us more insight in the process of interpreting multimodal data. We analyze the correspondences and differences between annotations for each case and we observe the complexity of the process and the unsuitability of using simple rules for fusion.

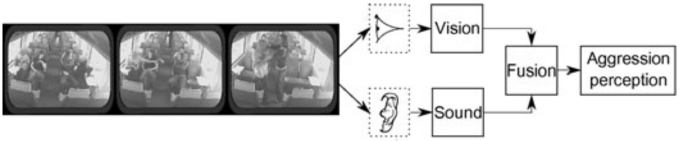


**Fig. 1.** Human model of aggression perception

We compare the performance of a detector trained on audio-only labels with one trained using multimodal labels. Using the unimodal annotation leads to approximately 10% absolute improvement. The same holds in the case of video. Having more accurate unimodal recognizers is bound to give a higher overall situation awareness and better results for multimodal fusion. In this work we restrict ourselves to low sensor features but the results can be improved by using higher level fusion, context or temporal effects.

This paper is organized as follows. In the next section we give an overview of related work. Next we describe our database of aggressive scenarios with details on the process of annotation and its results. We also provide an analysis of the annotation results. We continue with the details of the unimodal classifiers and the fusion strategies and results. The last section contains our conclusion.

## 2 Related Work

In [1] the influence of the individual modalities on the perception of emotions has been studied. One of their findings was that the agreement between annotators was the lowest for the multimodal case and the highest for audio only in the category based approach. For the dimensional approach, in the case of activation, it proved that strong emotions were present in all modalities, but for the lower active cases audio performed better than video.

Multimodal recognition of user states or of focus of attention of users interacting with a smart multimodal communication telephone booth was researched in [7]. In the case of user state recognition their results show that a second modality was reinforcing the clear cases but was not adding much gain in the doubtful cases. On the other hand, in the case of focus of attention, multimodality seemed to always help.

In [9] a smart surveillance system is presented, aimed at detecting instances of aggressive human behavior in public domains. At the lower level, independent analysis of the audio and video streams yields intermediate descriptors of a scene. At the higher level, a dynamic Bayesian network is used as a fusion mechanism that produces an aggregate aggression score for the current scene.

The optimal procedure of multimodal data annotation and the core mechanisms for multimodal fusion are still not solved. Furthermore, the findings seem to be dependent on the application [7]. Next, we analyze the issues related to multimodal annotation and recognition in the context of overt aggression.

# 3 Corpus of Multimodal Aggression

## 3.1 Database Description

We use the multimodal database described in [8]. The corpus contains recordings of semi-professional actors which were hired to perform aggressive scenarios in a train station setting. The actors were given scenario descriptions in terms of storyboards. In this way they are given a lot of freedom to play and interpret the scenarios and the outcomes are realistic. The frame rate for video is about 13 frames per second at a resolution of 640x256 pixels. Sound is recorded with a sample rate of 44100Hz with a 24 bit sample size.

We use 21 scenarios that span 43 minutes of recordings and are composed of different abnormal behaviors like harassment, hooligans, theft, begging, football supporters, medical emergency, traveling without ticket, irritation, passing through a crowd of people, rude behavior towards a mother with baby, invading personal space, entering the train with a ladder, mobile phone harassment, lost wallet, fight for using the public phone, mocking a disoriented foreign traveler and irritated people waiting at the counter or toilet.

## 3.2 Annotation

The annotation has been done in the following settings: *(i) audio-only* (the rater is listening to samples of the database without seeing the video), *(ii)video-only* (the rater is watching samples of the database without sound) and *(iii) multimodal* (the rater used both video and audio samples). For each annotation scheme the data has been split in segments of homogeneous aggression level by two of the annotators using Anvil [4].

For each segment we asked the raters to imagine that they are operators watching and / or listening to the data. They had to rate each segment on a 3 point scale as follows: *label 1* - normal situation, *label 2* - medium level of aggression / abnormality (the operator's attention is drawn by the data) and *label 3* - high level of aggression / abnormality (the operator feels the need to react). Besides the three point scale the annotators could choose the label 0 if the channel conveyed no information.

For each annotation setting the data was split in segments of homogeneous aggression level by two experienced annotators. In general, there was a finer segmentation for the audio - a mean duration of 8.5 seconds, and a coarser one for video and multimodal with mean segment durations of 15 and 16 seconds respectively. The different segment durations are inherent in the data and in the way each modality is dominant or not for a time interval. In the case of audio the resulting segment durations are shorter also because when people are taking turns to speak, the aggression level changes with the speaker.

Seven annotators rated the data for each setting (modality). The inter-rater agreement is computed in terms of Krippendorff's alpha for ordinal data. The highest value is achieved for audio, namely 0.77, while video and multimodal are almost the same, 0.62 and 0.63 respectively. One reason can be the finer segmentation that was achieved for audio, but also that raters perceived verbal aggression in very similar ways. The values do not reflect perfect agreement but are reasonable given the task.

Figure 2 displays distribution of the labels in terms of duration for each annotation setting. It can be noticed that the data is unbalanced with mostly neutral samples. However, the duration of the segments with label 3 is growing in the case of multimodal annotation. This can be caused by the additional context information that people get when using an extra modality and from the more accurate semantic interpretation that they can give to a scene, even if it does not look or sound extremely aggressive.

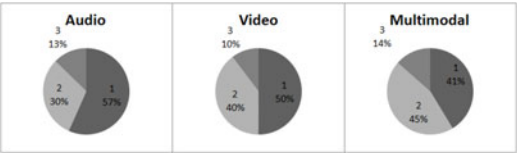


**Fig. 2.** The duration of each class based on the different annotations

## 4 Automatic Aggression Detection

In this section we describe in turn the audio, video and multimodal classifiers. Because we aim at an approach close to what we can expect in a real-life application and a real-time system, we refrain from using fine preset segmentations as turns in the case of audio or borders of actions in the case of video. Instead, we decide to base our analysis on segments of equal length (2 seconds). We expect that this choice leads to lower accuracies but the approach is the same as when we need to buffer data in a real-time detector.

In Figure 3 we summarize our results from the annotation and classification. The figure contains three types of values. The inner values represent the correlation coefficients between the annotations based on the three set-ups. The highest correlation is between the audio annotation and the multimodal one and the lowest is between the audio and the video annotations. This means that in many cases the audio and video annotations do not match, so we can expect that they provide complementary information which can be used for fusion. The connecting arrows between the inner and the outer figure represent the accuracies of the classifiers(in %). Each time the feature type matches the annotation type. The highest accuracy is obtained in the case of audio but we did not experiment yet with more advanced fusion methods that might improve the multimodal classifier. Finally, the values on the outer figure represent the correlations between the labels predicted by the classifiers. These values are lower than the values from the inner figure but in correspondance and reflect the results of the classifiers.

We have compared 3 classifiers on the audio features, video features and a concatenation of both. The classifiers are support vector machine (SVM) with a second order polynomial kernel, a logistic regression classifier, and AdaBoostM1. The differences between classifiers'performances are not significant, therefore we display only the best one for each case. The results using a 10 fold cross-validation scheme are presented in Table 2. For each case we report the true positive (TP) and false positive (FP) rates and the weighted averaged area under the ROC curve and display the confusion matrices in Table 3. Because the number of samples from each class is unbalanced, accuracy by itself is not a precise measure.

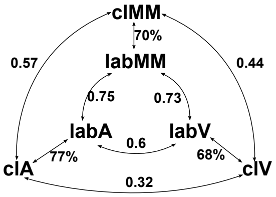


**Fig. 3.** Correlation coefficients between the original and predicted labels for the 3 modalities and the accuracies of the classifiers in %

## 4.1 Audio Processing

Vocal manifestations of aggression are dominated by negative emotions such as anger and fear, or stress. In the case of our audio recognizers we use a set of prosodic features inspired from the minimum required feature set proposed by [3] and [6]. The feature set consists of 30 features: speech duration, statistics (mean standard deviation, slope, range), over pitch (F0) and intensity, mean formants F1-F4 and their bandwidth, jitter, shimmer, high frequency energy, HNR, Hammarberg index, center of gravity and skew of the spectrum. In Table 1 we show the information gain for the best 20 features.

**Table 1.** Information gain for each feature

| Rank | Feature | Rank | Feature | Rank | Feature | Rank | Feature |
|---|---|---|---|---|---|---|---|
| 0.423 | mean I | 0.287 | bw1 | 0.216 | bw2 | 0.161 | mean F0 |
| 0.409 | high energy | 0.284 | HF500 | 0.198 | slopeltaspc | 0.152 | HNR |
| 0.387 | max I | 0.25 | range I | 0.197 | mean F3 | 0.144 | shimmer |
| 0.379 | slope I | 0.218 | skew S | 0.184 | cog S | 0.132 | HF1000 |
| 0.294 | bw3 | 0.216 | std I | 0.161 | max F0 | 0.127 | duration |

Difficulties of processing realistic data arise from high and different levels of noise. As a first preprocessing step we use a single-channel noise reduction algorithm based on spectral subtraction. We used the noise reduction scheme used in [2] in combination with the noise PSD tracker proposed in [2]. This procedure solves the noise problem but musical noise artifacts appear, which generate additional F0. Since solving musical noise is a complex problem and out of our scope we decided to use the original samples. Another problem inherent in naturalistic data is the existence of overlapping speech. Because of overlapping speech and the fixed sized unit segment, the F0 related features have a lower information gain than expected.

### 4.2 Video Processing

For the video processing unit we use the approach in [5]. The video segments are described in terms of space-time interest points (STIP) which are computed for a fixed set of multiple spatio-temporal scales. For the patch corresponding to each interest point, two types of descriptors are computed: histograms of oriented gradient (HOG) and histograms of optical flow (HOF). These descriptors are used based on a bag-of-words approach. A codebook is computed using the k-means algorithm and each feature vector is assigned to a visual word based on the smallest Euclidean distance. For each time segment that we analyze we compute histograms which compose the final feature set for the classifier.

We have tested this approach with vocabularies of different sizes, with HOG, HOF and HNF (a concatenation of HOG and HOF) and with unit length of 1,2 and 3 seconds. The best results were obtained for HNF using a vocabulary of 30 words and for time segments of 2 seconds and the AdaBoostM1 classifier (see Table 2).

### 4.3 Multimodal Processing

The feature vectors from audio and video are concatenated into a new multimodal feature vector that becomes the input of our classifier (using the multimodal labels), an approach known as feature-level fusion. The performance is in between the performances of audio only and video only.

### 4.4 Results

As expected, the accuracy of the unimodal recognizers increased when using the unimodal labels as ground truth. The improvement is as high as 11% absolute in the case of audio and 9% absolute in the case of video, as can be noticed from Table 2. We realize that a more advanced fusion will result in better performance but this is a basic approach.

**Table 2.** Results for audio recognizers with audio-only and multimodal ground truth

| Features | Labels | Classifier | TP | FP | ROC |
|---|---|---|---|---|---|
| A | A | SVM | 0.77 | 0.19 | 0,81 |
| | MM | | 0.66 | 0.25 | 0,74 |
| V | V | AdaBoostM1 | 0.68 | 0.26 | 0.77 |
| | MM | | 0.59 | 0.30 | 0.72 |
| MM | MM | AdaBoostM1 | 0.70 | 0.22 | 0.84 |

### 4.5 Unimodal Differences and Consequences for Multimodal Classification

The unimodal annotations allow us to have more insight into how multimodality works. In many cases the labels from audio, video and multimodal annotation match, but there are also many cased when they do not. The examples presented in Table 4 can give a hint of how complex the fusion process is, hence we can not expect to mimic it by simple rules or classifiers.

**Table 3.** Confusion matrices for: audio classifier(left), video classifier(middle), multimodal classifier(right)

| Classified as | | | |
|---|---|---|---|
| **1** | **2** | **3** | Correct |
| **629** | 76 | 4 | **1** |
| 133 | **261** | 22 | **2** |
| 6 | 53 | **94** | **3** |

| Classified as | | | |
|---|---|---|---|
| **1** | **2** | **3** | Correct |
| **490** | 132 | 9 | **1** |
| 190 | **321** | 8 | **2** |
| 20 | 53 | **51** | **3** |

| Classified as | | | |
|---|---|---|---|
| **1** | **2** | **3** | Correct |
| **401** | 116 | 1 | **1** |
| 127 | **439** | 22 | **2** |
| 14 | 97 | **54** | **3** |

**Table 4.** Example of scenes with different multimodal scores. The abbreviations stand for: A=audio, V=video, MM=multimodal, H=history, C=context, S=semantics, W=words.

| A | V | MM | Scene | Problem type |
|---|---|---|---|---|
| 2 | 2 | 3 | touching | S,W |
| 3 | 2 | 3 | verbal fight | H |
| 1 | 2 | 1 | funny non aggressive movement | S |
| 2 | 3 | 2 | aggressive movement but it does not sound serious | S |
| 1 | 3 | 3 | physical fight, silent | H,S,C |
| 3 | 3 | 2 | person pushing through crowd, people complaining | H,S,C |
| 1 | 3 | 2 | person accused of stealing wallet, sounds calm, movement | H,S,C |

## 5 Conclusions and Future Work

In this paper we have approached the problem of multimodal aggression detection based on the human model of perception. Our results show that using an extra level of annotations improves the recognition of the classifiers with on average 10%. The accuracies are still far from 100%, but we are only using very simple features, with no semantic interpretation, no history and no context. For example, for audio we analyse how something is said but we do not take into account the meaning of words. In the future we will use speech recognition in order to find key words and have more insight into the meaning. In the case of video we currently employ movement features. Nevertheless, a lot of aggression can be seen from facial expression, body gestures, aggressive postures, which are not yet incorporated in our system.

Another benefit of analyzing data both unimodal and multimodal is that we have insight into the complexity of the problem. Applying a simple rule or a classifier to solve the fusion problem is not sufficient. In our future work we plan to use a reasoning based approach and probabilistic fusion including reasoning over time with dynamic Bayesian networks.

## References

1. Douglas-Cowie, E., Devillers, L., Martin, J.C., Cowie, R., Savvidou, S., Abrilian, S., Cox, C.: Multimodal databases of everyday emotion: Facing up to complexity. In: Ninth European Conference on Speech Communication and Technology (2005)

2. Hendriks, R.C., Heusdens, R., Jensen, J.: MMSE based noise PSD tracking with low complexity. In: IEEE Int. Conf. Acoust, Speech, Signal Processing, pp. 4266–4269 (2010)
3. 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)
4. Kipp, M.: Spatiotemporal Coding in ANVIL. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008 (2008)
5. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE Int. Conf. of Computer Vision and Pattern Recognition (2008)
6. Lefter, I., Rothkrantz, L.J.M., Wiggers, P., Van Leeuwen, D.A.: Emotion recognition from speech by combining databases and fusion of classifiers. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 353–360. Springer, Heidelberg (2010)
7. Nöth, E., Hacker, C., Batliner, A.: Does multimodality really help? The classification of emotion and of On/Off-focus in multimodal dialogues-two case studies. In: ELMAR, pp. 9–16 (2007)
8. Yang, Z.: Multi-Modal Aggression Detection in Trains. PhD thesis, Delft Univeristy of Technology (2009)
9. Zajdel, W., Krijnders, J.D., Andringa, T.C., Gavrila, D.M.: CASSANDRA: audio-video sensor fusion for aggression detection. In: Proc. IEEE Conference on Advanced Video and Signal Based Surveillance AVSS, pp. 200–205 (2007)

# Analysis of Data Collected in Listening Tests for the Purpose of 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.** In this paper we present an analysis of data, which were collected in listening tests, and are planned to be used for the development and evaluation of concatenation cost functions for unit selection based TTS systems. The aim of the analysis was to evaluate a "richness" of the collected data with respect to the intended utilization. No effort was made to propose a new method for measuring concatenation artifacts. The study was limited to two speakers (male and female), and five short 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:** speech synthesis, speech analysis, unit selection, concatenation cost, data collection.

## 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 [1].

In order to select the best sequence of units from the available database, two cost functions are typically calculated—*target cost function* and *concatenation (join) cost function*. 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 mostly consists of sub-components associated with a difference in pitch, energy and spectral envelops of adjacent segments of concatenated units. The spectral component is considered to be a weak point as no objective measure seems to correlate well with human perception of discontinuities in spectrum.

[⋆] This research was supported by the Grant Agency of the Czech Republic, project No. GAČR 102/09/0989; by the Technology Agency of the Czech Republic, project No. TA01011264. The work has also been supported by the grant of the University of West Bohemia, project No. SGS-2010-054.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 33–40, 2011.

A large number of methods have been proposed and/or evaluated over last one and a half decades ([2], [3], [4], [5], [6]), but none of them proved to be comparatively better than others. The presented results have sometimes even been in contradiction. Despite the activities in this direction tend to be fading out, the design of a concatenation cost function reliably reflecting human perception is still an open issue.

In order to assess and/or design concatenation cost functions, one needs to collect rich enough perceptually evaluated data, which would contain large spectrum of different kinds of discontinuities as well as smooth non–disturbing concatenations. The purpose of this paper is to present an analysis of data we have collected in listening tests following the recommendations given in [7], and to discuss the findings of the analysis. Whereas our previous work [8] was more focused on the analysis of listeners' ratings, here we put stress on the content of the collected data.

## 2 Perceptual Data Collection

### 2.1 Sentence Material

Recordings covering five Czech short vowels in all consonantal contexts were made in an anechoic room by two professional speakers—male and female. The recorded scripts were composed of three word sentences containing consonant-vowel-consonant (CVC) word in the middle each, e.g. `/kra:lofski:` **`kat`** `konal/` (Czech SAMPA notation). Recorded data were re-synthesized using the "half sentence" method [7]. This method consists in cutting the sentences in the middle of the vowels in the central words, and combining the left and right parts, which results in a large set of sentences containing only one concatenation point in the middle of the central CVC word each.

### 2.2 Sentence Selection Considerations

Since the amount of synthesized sentences was too big to be entirely used as listening tests stimuli, a selection mechanism was needed. We have learned from our preliminary study [7] that differences in pitch and energy at concatenation points are important factors to be taken into consideration. In that study, all perceptually discontinuous concatenation points were found to be clearly separable from the continuous ones in the $F0$ difference × Energy difference plane (henceforth referred to as $\Delta F0 \times \Delta En$ plane, see Fig. 2), which is not useful for evaluating the spectral component of the concatenation cost functions.

In most related studies listed above, different smoothing and/or restrictive procedures were applied to ensure that audible discontinuities at the concatenation points are not due to pitch or energy differences. Since we did concatenations without applying any smoothing methods to limit a risk of introducing audible signal degradations, which could influence the listeners' ratings, we decided to calculate the differences in $F0$ and energy at concatenation points and take them into account when selecting the listening tests stimuli sentences.

### 2.3 Selection Methods

Depending on the selection methods, the listening tests stimuli sentences can be divided into sets summarized in Tab. 1. The sets f0B, enB and efB were included to confirm that large differences in pitch and energy at concatenation points are significant sources 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 in which the difference in pitch was less than 10 mels were used. The sets efB and efS were composed of sentences with the largest and smallest Euclidean distance from the origin in the $\Delta F0 \times \Delta En$ plane, respectively.

The pitch and energy differences were calculated from one pitch period on either side of concatenation points, having all recordings previously pitch marked using the Multi-Phase Pitch-Mark Detection Algorithm [9].

Based on the results presented in [7], we put more stress on sentences with smooth pitch and energy transitions at concatenation points, i.e. the sets mfS, beS, mfB, beB. The selection of candidates to be included to these sets was based on ranking all the synthesized sentences according to the Euclidean distances from the origin in the $\Delta F0 \times \Delta En$ plane and taking into consideration only 33% best ones. Then, spectrum oriented methods described below were applied.

The mel-frequency cepstral coefficients (MFCCs) based distance ($\Delta MFCC$) was calculated as the Euclidean distance between two standard 39-dimensional MFCCs vectors characterizing the left and right one pitch period long segments of the boundary region, respectively. The calculation of the latent semantic mapping (LSM) based distance ($\Delta LSM$) was done in line with the method presented in [5]. The dimension of the singular value decomposition was set to 10 and the length of the extraction window was set as K=3.

All selected sentences have been checked manually to ensure that they are phonetic segmentation and pitch marking errors free. The total number of sentences presented to listeners in each listening test was 1310.

**Table 1.** Sets of sentences contained in the listening tests stimuli. The symbols $\Uparrow \Delta$ and $\Downarrow \Delta$ stand for a "large difference" and a "small difference", respectively.

| Set | Description | Num. |
|---|---|---|
| f0B | $\Uparrow \Delta F0$ $\Downarrow \Delta En$ | 150 |
| enB | $\Uparrow \Delta En$ $\Downarrow \Delta F0$ | 150 |
| efB | $\Uparrow \Delta F0$ $\Uparrow \Delta En$ | 150 |
| efS | $\Downarrow \Delta F0$ $\Downarrow \Delta En$ | 75 |
| mfS | $\Downarrow \Delta F0$ $\Downarrow \Delta En$ $\Downarrow \Delta MFCC$ | 75 |
| beS | $\Downarrow \Delta F0$ $\Downarrow \Delta En$ $\Downarrow \Delta LSM$ | 75 |
| mfB | $\Downarrow \Delta F0$ $\Downarrow \Delta En$ $\Uparrow \Delta MFCC$ | 225 |
| beB | $\Downarrow \Delta F0$ $\Downarrow \Delta En$ $\Uparrow \Delta LSM$ | 225 |
| ran | random selection | 135 |
| nat | original recordings | 15 |
| rev | revision sentences | 15 |
| dbl | same source and target context | 20 |

### 2.4 Listening Tests Subjects

The subjects were university students, all native speakers of Czech. A few listeners stated that they had some background in phonetics. There were 29 subjects who finished the first listening test (male voice) and 27 subjects 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.

### 2.5 Listening Tests Procedure

The task of the listeners was to assess the concatenations on both a five-point scale (*no join at all*, *unnatural but not disturbing join*, *slightly perceived join*, *highly perceived join* and *highly disturbing join*) 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 tests instructions that the tests shall be done in a silent environment and using headphones. 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 [8]. To help the listeners calibrate for the more fine grained scale, a preparation phase was included containing various examples of audible discontinuities. It was allowed to listen to the calibration sentences at any time during the listening tests. There were no restrictions on how many times the listeners played each sentence before assessing it.

## 3 Data Analysis

### 3.1 Analysis of Listeners' Answers

In order to identify listeners who did not show good agreement with the majority, a rigorous analysis of the listeners' ratings has been performed [8]. We ranked the participants according to the scores obtained by the analysis, and 9 and 6 participants were excluded from the male and female voice listening tests, respectively. The results presented in the following sections do not contain ratings given by those listeners.

### 3.2 Collection of "facts"

The next step of the analysis was to collect two sets of "facts", i.e. sentences that were assessed by more than or equal to 80% of listeners in the same way on the binary scale. The set of "facts" can be formally described as:

$$\mathrm{sent}_i \in \mathrm{FACTS} \quad \Leftrightarrow \quad \frac{\mathrm{N}_i^+}{\mathrm{N}_i} \geq 0.8 \;\vee\; \frac{\mathrm{N}_i^-}{\mathrm{N}_i} \geq 0.8, \tag{1}$$

where $\mathrm{sent}_i$ is the $i$-th sentence of the test stimuli, FACTS stands for the set of "facts", $\mathrm{N}_i^+$, $\mathrm{N}_i^-$ are the numbers of continuous (i.e. *not perceived join*) and discontinuous (i.e.

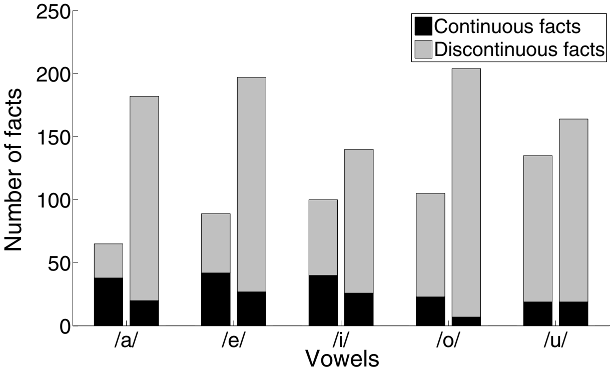


**Fig. 1.** The "facts" collected in the listening tests sorted by the vowels—the left bar in each pair represents the male voice results

*perceived join*) ratings given to the $i$-th sentence, respectively, and $\mathrm{N}_i$ is a total number of ratings given to the $i$-th sentence.

Results of the collection of "facts" are given, sorted by the vowels, in Fig. 1. For the male voice, balanced facts distributions were obtained for the vowels /a/,/e/ and /i/. The number of audible discontinuities was comparatively higher for the vowels /u/ (in line with the results presented in [4]) and /o/. For the female voice, more "facts" were collected for all vowels, which is mainly attributed to the discontinuous "facts", especially for the vowels /a/,/e/ and /o/.

The total number of collected "facts" was 494 for the male voice and 887 for the female voice. These numbers imply that for the male voice, more sentences fell within a "gray" area, where the listeners were not sure about their ratings using the binary scale.

### 3.3 Distribution of "facts" in $\Delta F0 \times \Delta En$ Plane

The primary goal of the design of the listening tests was to collect perceptual data exploitable for measuring spectral discontinuities, i.e. having continuous and discontinuous "facts" mixed when displayed in the $\Delta F0 \times \Delta En$ plane. Nevertheless, it was also interesting to see how big difference in pitch and energy at concatenation points is allowed without being perceived by the listeners.

The distributions of continuous and discontinuous "facts" for all voice/vowel combinations looked similarly to the one given in Fig. 2 showing the male voice vowel /e/. Interestingly, there have been some sentences which were assessed as continuous "facts" in spite of containing relatively large differences in pitch and energy at concatenation points. It is noteworthy that the plot shows only static analysis not covering dynamics and slopes of the $F0$ contours, which may also play an important role in the perception of discontinuities.

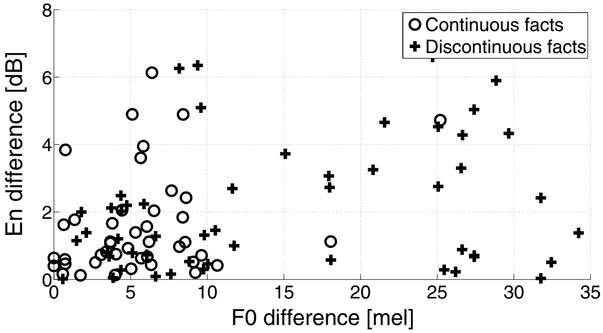


**Fig. 2.** Distribution of the "facts" in the $\Delta F0 \times \Delta En$ plane—vowel /e/, male voice

### 3.4 Distribution of "facts" in Test Stimuli Sets

As described in Sec. 2.3, various methods listed in Tab. 1 were used to select listening tests stimuli sentences. The motivation for using those methods was to gain control, albeit limited, over the listening test results (in terms of "facts" counts) without having any a priori knowledge about the distribution of audible discontinuities in the synthesized data.

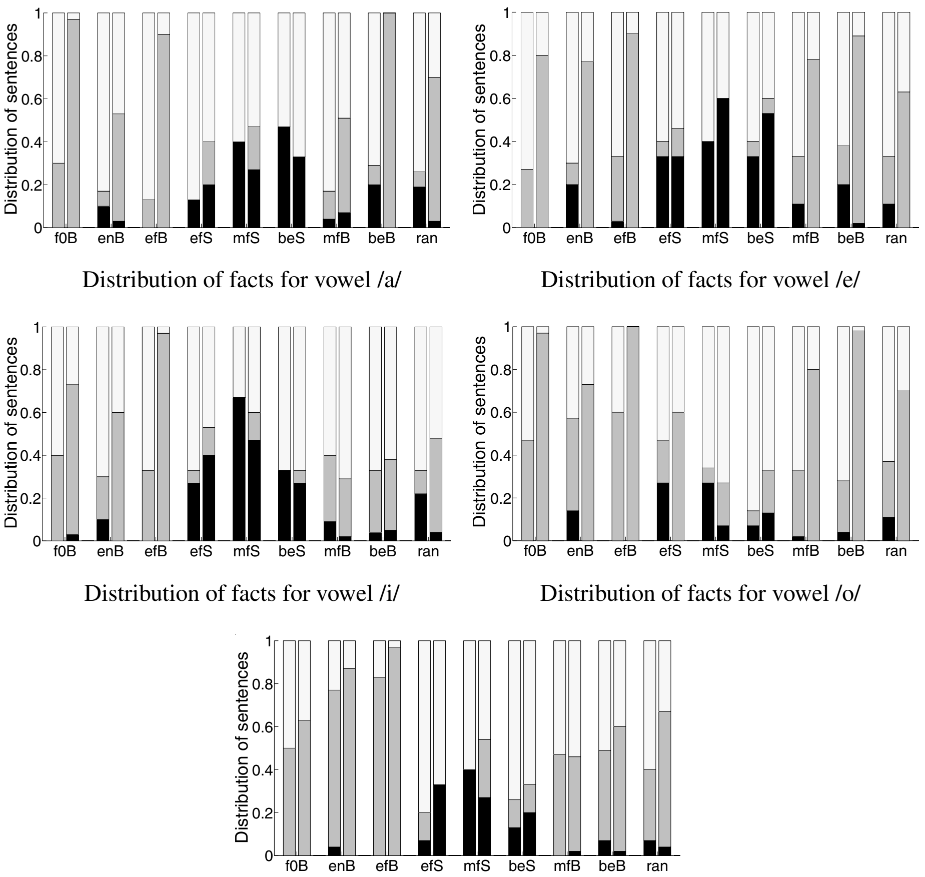
The sets `f0B`, `enB` and `efB` were expected to lead to discontinuous "facts", the sets `efS`, `mfS`, `beS`, `nat` and `dbl` to continuous "facts", and the sets `mfB` and `beB` rather to discontinuous "facts". The obtained results given in Fig. 3 sorted by the vowels and the sets confirm that we only gained limited control by introducing the selection methods. There are, however, some remarkable observations.

Upon a closer inspection of the `ran` sets for all vowels, one can see that the female voice data contained considerably larger amount of audible continuities disregarding any selection method. Similar finding has already been presented in other studies comparing male and female voices.

The sets `f0B`, `enB`, `efB`, `mfB` and `beB` show for the vowels /a/, /e/ and /o/ significantly smaller number of the discontinuous "facts" for the male voice. In contrast, none of the sets have shown different distributions for the vowel /u/, and the same can be said about the sets `mfB` and `beB` for the vowel /i/.

If we turn next to the continuous "facts", it can be seen that to find any pattern across all vowels and sets is rather difficult. The differences in their quantities across the sets overall compensate resulting in the comparable counts across the two voices as shown in Fig. 1. The only exception is the vowel /o/, for which surprisingly large number of discontinuous facts can be found in the set `efS` for the female voice.

Another observation that is worth mentioning when comparing the two voices is for the `beB` sets. Specifically, these sets generated very different results for the vowels /a/, /e/ and /o/, for which our expectation was nearly perfectly met for the female voice, in contrast to the male voice.



Distribution of facts for vowel /a/

Distribution of facts for vowel /e/

Distribution of facts for vowel /i/

Distribution of facts for vowel /o/

Distribution of facts for vowel /u/

**Fig. 3.** Distributions of the "facts" collected in both listening tests sorted by the test stimuli sets—left bar in each pair represents the results for the male voice. Continuous "facts" are represented by black color, gray is used to show the proportion of discontinuous "facts".

## 4 Conclusions and Future Work

This paper described the analysis of data collected in the listening tests for the purpose of the evaluation of the concatenation cost functions. The aim of the performed analysis was to examine the "richness" of the collected data with respect to the intended future utilization. The evaluation of the performance of different discontinuity measures was not the main objective of this work.

It was found that despite using the same methods for the selection of synthetic sentences included into the listening tests, considerably larger number of audible discontinuities was found for the female voice. The obtained results were only comparable across the two voices for the vowel /u/. For the male voice, the collected data are nicely balanced (in terms of continuous and discontinuous joins) for the vowels /a/, /e/ and /i/, but also for the other two vowels the data are still useful. The results collected for the vowel /o/ have been disappointing as not enough continuous concatenations were found for the female voice.

As a next step, it is planned to closely inspect the reasons of the larger number of audible discontinuities observed for the female voice as well as to analyze the role of $F0$ slopes at concatenation points in the perception of discontinuities. We also want to more closely inspect the sentences containing large $F0$ and energy differences at concatenation points without being perceived by listeners.

## References

1. Hunt, A., Black, A.: Unit selection in a concatenative speech synthesis system using a large speech database. In: ICASSP 1996, Atlanta, Georgia, vol. 1, pp. 373–376 (1996)
2. Vepa, J.: Join cost for unit selection speech synthesis. Ph.D. thesis, University of Edinburgh (2004)
3. Stylianou, Y., Syrdal, A.K.: Perceptual and objective detection of discontinuities in concatenative speech synthesis. In: ICASSP 2001, Salt Lake City, Utah, vol. 2, pp. 837–840 (2001)
4. Klabbers, E., Veldhuis, R.: Reducing audible spectral discontinuities. IEEE Transactions on Speech and Audio Processing 9, 39–51 (2001)
5. Bellegarda, J.R.: A novel discontinuity metric for unit selection text-to-speech synthesis. In: SSW5 2004, Pittsburgh, PA, USA, pp. 133–138 (2004)
6. Pantazis, Y., Stylianou, Y.: On the detection of discontinuities in concatenative speech synthesis. In: Stylianou, Y., Faundez-Zanuy, M., Esposito, A. (eds.) COST 277. LNCS, vol. 4391, pp. 89–100. Springer, Heidelberg (2007)
7. 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)
8. Legát, M., Matoušek, J.: Collection and analysis of data for evaluation of concatenation cost functions. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 345–352. Springer, Heidelberg (2010)
9. Legát, M., Matoušek, J., Tihelka, D.: A robust multi-phase pitch-mark detection algorithm. In: INTERSPEECH 2007, Antwerp, Belgium, vol. 1, pp. 1641–1644 (2007)

# Analysis of Inconsistencies in Cross-Lingual Automatic ToBI Tonal Accent Labeling

David Escudero-Mancebo, Carlos Vivaracho Pascual, César González Ferreras, Valentín Cardeñoso-Payo[1], and Lourdes Aguilar[2]

[1] Dpt. of Computer Sciences, Universidad de Valladolid, Spain
[2] Dpt. of Dpt. of Spanish Philology, Universidad Autónoma de Barcelona, Spain

**Abstract.** This paper presents an experimental study on how corpus-based automatic prosodic information labeling can be transferred from a source language to a different target language. Tone accent identification models trained for Spanish, using the ESMA corpus, are used to automatically assign tonal accent ToBI labels on the (English) Boston Radio news corpus, and vice versa. Using just local raw prosodic acoustic features, we got about 75% correct annotation rates, which provides a good starting point to speed up automatic prosodic labeling of new unlabeled corpora. Despite the different ranges and relevance of inter corpora acoustic input features, the contrasting of the results with respect to manual labeling profiles indicate the potential capabilities of the procedure.

**Index Terms:** prosodic labeling, cross-lingual prosody, automatic accent identification.

## 1 Introduction

ToBI has been implemented for several languages including English, German and Japanese. Despite the intensive research activity for Iberian languages, the need of a reference corpus similar to those existing for other languages (e.g. the Boston Radio Corpus for English [9]) is still a need both for Catalan and Spanish. The activity presented in this paper is included in the Glissando project[1], which aims to record and label a bilingual Spanish and Catalan corpus that contains Radio news recordings and spontaneous dialogs with ToBI marks [11]

Labeling a corpus with ToBI tags is an expensive procedure. In [12] it is estimated that the ToBI labeling commonly takes from 100-200 times the real time. To speed up the process, automatic or semiautomatic methods would seem to be a productive resource. [2] or [10] are good examples of the state of the art on automatic labeling of ToBI events. For Catalan, [1] presents a procedure to label break indices, reducing the set of break indices by merging some of them together in order to increase the identification results. This merging strategy is common in other studies such as the ones already mentioned from [2] or [10] that combine the different types of accent tones, transforming the labeling problem into a binary one in order to decide whether

[1] Partially funded by the Ministerio de Ciencia e Innovacion, Spanish Government Glissando project FFI2008-04982-C003-02 and Junta de Castilla y León JCYL2011.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 41–48, 2011.

**Table 1.** Number of words in the corpora and subcorpora

| | Words | | | Syllables | | |
|---|---|---|---|---|---|---|
| | #total | #accented | #un-accented | #total | #accented | #un-accented |
| ESMA-UPC | 7236 | 2483 | 4753 | 14963 | 2341 | 12622 |
| BURNC | 27767 | 13899 | 13868 | 47323 | 14873 | 32450 |

an accent is present or not. In this paper an automatic labeling procedure of accent tones (binary decision) is presented that aims to speed up the job of the manual labelers who will be required to check the predictions of the system.

Here we explore a cross-lingual approach where a given corpus with ToBI labels will be used to predict the labels of a different corpus in a different language. Despite the fact that the ToBI sequences are highly dependent on the language, they codify universal functions of prosody, including the prominence (here prominence is identified with the presence of a ToBI accent tone mark). Thus, the Boston Radio Corpus is used to train prosodic models that are then used to identify the presence of tonal accent in a Spanish corpus and vice versa. Results are promising as, using raw prosodic features, close to 75% of the words are correctly classified.

This cross-lingual approach is thus an opportunity for prosodic studies, as the number of linguistic resources with ToBI labels is sparse and the number of languages that lack this information is still large. At the same time, we assume that there are challenges to cope with, so differences in the relevance and scale of the input features in the different languages are analyzed and identified as the battlefield on wich to increase the performance of the classifier in future works. First, the experimental procedure is presented followed by the results on cross-lingual accent identification. Discussion and future work end the paper.

## 2 Experimental Setup

The cross-lingual approach does not solely consist of training classification models with the data of a corpus in a given language in order to test with the data of a different one. Additionally, we systematically contrast the differences on the input features among the diverse corpora in several practical aspects. First, the scale of the input features is analyzed to contrast the differences among the languages and the impact of the normalization of the input is shown.

The cross-lingual study continues with the examination of the relevance of the input features in the different corpora. Input features are ranked in terms of theirs informative capabilities for discriminating whether a word or a syllable is accented. Every language has their own ranking to be contrasted. Furthermore, the most informative input features are also analyzed to verify whether significant differences appear among the diverse corpora (e.g. f0_range of the accents in Spanish vs. accents in English).

Finally, automatic prediction results are contrasted with perceptual judgements made by a set of labelers on the same testing corpus. This is useful to value the usefulness of the automatic labeling process in general and of the cross-lingual labeling in particular.

**Table 2.** Classification rates (in percentages) using words in terms of the presence of accent. The training/testing corpora in the rows. C4.5 is the decision tree; MLP is the neural network refering to each type of classifier used. *NI* in normalized input. *Ov* is oversampled input. *T* is total, *A* is accented and *U* is unaccented.

| Corpus | C4.5$(\bar{NI})(\bar{Ov})$ | | | C4.5$(NI)(\bar{Ov})$ | | | C4.5 $(\bar{NI})(Ov)$ | | | C4.5$(NI)(Ov)$ | | | MLP(NI)(Ov) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Trainning/Testing | T | A | U | T | A | U | T | A | U | T | A | U | T | A | U |
| ESMA-UPC/ESMA-UPC | 79.6 | 64.0 | 87.7 | 79.6 | 64.0 | 87.7 | 79.6 | 81.2 | 77.8 | 78.7 | 81.9 | 75.3 | 77.0 | 75.7 | 77.6 |
| ESMA-UPC/BURNC | 65.1 | 36.5 | 93.8 | 71.2 | 55.4 | 87.1 | 68.3 | 46.4 | 90.3 | 73.4 | 67.8 | 79.0 | 72.6 | 67.5 | 77.8 |
| BURNC/ESMA-UPC | 61.8 | 84.1 | 50.2 | 73.2 | 78.4 | 70.5 | 68.8 | 86.5 | 50.2 | 74.8 | 78.9 | 70.5 | 66.6 | 81.1 | 59.0 |
| BURNC/BURNC | 77.0 | 75.6 | 78.4 | 77.6 | 78.2 | 77.1 | 77.0 | 75.6 | 78.4 | 77.6 | 78.2 | 77.1 | 78.8 | 80.1 | 77.5 |

### 2.1 Processing of the Corpora

We used the Boston University Radio News Corpus BURNC[9]. This corpus includes labels separating phonemes, syllables and words. Accents are marked with a ToBI label and a position. Inspired in state of the art works [14,2], the accent tones were aligned with respect to the prominent syllable and the word containing it (table 1).

The Spanish corpus used in this paper is ESMA-UPC. It was designed for the construction of a unit concatenative TTS system for Spanish [3]. Although it was not specifically designed for prosodic studies, it contains enough data to get significant results. Speech was recorded in one channel and the output of a laryngograph in another one. Data were automatically labeled and manually supervised. Labeling included silences, allophonic transcription, and allophonic boundaries. This information was increased by the additional syllable and word boundaries and accent positions(table 1).

Feature extraction in both corpora was carried out using similar features to other experiments reported in the bibliography [2]. The features concern frequency: within word F0 range (f0_range), difference between maximum and average within word F0 (f0_maxavg_diff), difference between average and minimum within word F0 (f0_minavg_diff), difference between within word F0 average and utterance average F0 (f0_avgutt_diff); energy: within word energy range (*e_range*), difference between maximum and average within word energy (e_maxavg_range), difference between average and minimum within word energy (e_minavg_range); duration: maximum normalized vowel nucleus duration from all the vowels of the word (normalization is done for each vowel type) (duration).

Although POS Tagging information and context have shown to be helpful in the improvement of the classification results (see [14,2]), this information was not used in the experiments reported in this paper. There is no obvious correspondence between POS tags used in each corpus: BURNC corpus uses the Penn Treebank tag set[8] and ESMA uses the EAGLES tag set for Spanish labeled using the Freeling tagger (`http://www.freeling.org`). A valid correspondence is under study for its application in future works. Regarding to context, we focus on local effects (at the level of word and/or syllable) as the context can be highly dependent on the language and the modeling of its correspondence is beyond the scope of this paper.

### 2.2 The Classifiers

A Multilayer Perceptron (MLP) with a non-linear sigmoid unit is trained for each classification problem, using the Error Backpropagation learning algorithm. Several network configurations were tested, achieving the best results with the following: i) single hidden layer with 12 neurons (more hidden units than inputs were used to achieve separation surfaces between closed classes), ii) 100 training epochs, iii) two neurons in the output layer, one for each class to be classified, then the test input vector is assigned to the class corresponding to the largest output.

The Weka machine learning toolkit [7] was used to build C4.5 decision trees (J48 in Weka). Different values for the confidence threshold for pruning have been tested, although the best results are obtained with the default value (0.25). The minimum number of instances per leaf is also set to the default value (2).

In [6] the negative impact of imbalanced data on final result is shown. Therefore, re-sampling methods are applied: minority class example repetition [13] for the MLP classifier and Synthetic Minority Oversampling TEchnique (SMOTE) method [4] for the C4.5 classifier. Due to the different scale of the features among the training corpora, we applied different normalization techniques: the Z-Norm, Min-Max, divide by maximum and euclidean norm 1. The normalization has been processed by corpus and by speacker.

## 3 Results

In spite of the fact that input features are relative magnitudes (differences with respect to a mean value), significant differences appear between the diverse corpora. These differences were expected, independently of the cross-lingual effect, as the different recording conditions have a clear impact on the values of the input magnitudes. Thus, for example, the ESMA F0 values have been collected with a laringograph device and BURNC F0 values with a pitch tracking algorithm leading to less stable values. Differences between the Spanish corpus and the English one are clear, so that the normalization of the input features is thus a need in this work.

The impact of the normalization is clearly seen in table 2 (columns $NI$ vs. $\bar{NI}$): results significantly improve when the input is normalized in the cross-corpus scenarios. Oversampling the corpora ($Ov$ vs $\bar{O}v$ columns) has also a positive impact to reduce the imbalanced results corresponding the accented vs. unaccented classes. In the conventional scenarios (same corpus for training and testing) results are obtained going up to 79.6%, which are the expected results according to the state of the art: [10] reports state of the art up-to-date results from 75.0% to 87.7% using the Boston Radio Corpus but adding the morpho-syntactic POS tag information and taking into account the context. In the cross-lingual scenarios, the classification rates decrease but they get to acceptable results taking into account that a posterior manual revision of the predictions will be applied. Syllables are also used as the reference unit with similar results (words in tables 2).

In order to analyze the reasons for the performance decrease in the cross-lingual accent classification task, we contrast the informative capabilities of the input features in the Spanish and English corpora. Table 3 compares the *Information Gain* of the different features, providing a measure of the potential loss of entropy which would be generated if the splitting of the training set was carried out in terms of the present feature [15]. The tagging of the Spanish corpus seems to rely mainly on F0 features, as

**Table 3.** Info Gain (IG), computed with the WEKA software, of the features when they are used to classify the accents in the different corpora

| ESMA-UPC | | BURNC | |
|---|---|---|---|
| Feature | IG | Feature | IG |
| f0_minavg_diff | 0.18888 | f0_minavg_diff | 0.248 |
| f0_range | 0.18246 | f0_range | 0.231 |
| f0_avgutt_diff | 0.15215 | f0_maxavg_diff | 0.191 |
| f0_maxavg_diff | 0.10891 | e_range | 0.184 |
| e_range | 0.09695 | e_maxavg_diff | 0.162 |
| e_minavg_diff | 0.08156 | duration | 0.159 |
| e_maxavg_diff | 0.07681 | e_minavg_diff | 0.141 |
| duration | 0.0063 | f0_avgutt_diff | 0.121 |

**Table 4.** Statistics of the input feature f0_minavg_diff

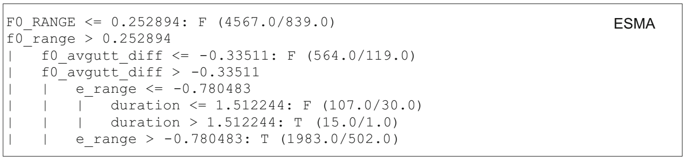
| | ESMA-UPC | BURNC |
|---|---|---|
| Accented data: | mean=0.67 sd=1.00 | mean=0.48 sd=1.07 |
| Unaccented data: | mean=-0.35 sd=0.80 | mean=-0.49 sd=0.62 |
| Total: | mean=0.0 sd=1.0 | mean=0.0 sd=1.0 |

the four most relevant features are related with F0 and the difference with respect to the energy and duration features is important. The tagging of the English corpus also seems to rely mainly on F0 features ($f0\_minavg\_diff$ and $f0\_range$ share the top ranking position in both corpora). Nevertheless, differences appear as the energy and duration appear to be more relevant for the English transcribers than for the Spanish ones.

Table 4 shows the discrimination capabilities of the most informative input feature (f0_minavg_diff) for the ESMA and the BURNC corpora. There are no statistical differences for this variable between the Spanish and English corpora (see the row that has the title *Total* in the table 4). Nevertheless, when the data are split into the classes, statistically significant differences clearly appear between the accented words of the diverse corpora and between the unaccented wordds (t-test p-value=2.2e-16). These differences justify the performance decrease of the classification task in cross-lingual situations observed in table 2.

## 4 Contrasting Automatic and Manual Labeling

One of the advantages of using automatic classifiers in contrast with manual labeling is the introduction of objective criteria. In [5] an experiment was presented where a subset of 20 utterances from the ESMA corpus were manually labeled by a team of six ToBI human raters. The inter-transcriber agreement indicated that there are two groups of labelers using two potential labeling criteria. In group one the inter-pair agreement goes from 86.7 to 92.0% while agreement of the 2 labelers of the second group is 94.7%. The inter group agreement decreases, going down to the range 65.5 to 73.5%. We hypothesize here that the availability of automatic prediction is an opportunity not only to speed up the manual labeling process, but also to offer an objective criteria to



```
F0_RANGE <= 0.252894: F (4567.0/839.0)                                    ESMA
f0_range > 0.252894
|   f0_avgutt_diff <= -0.33511: F (564.0/119.0)
|   f0_avgutt_diff > -0.33511
|   |   e_range <= -0.780483
|   |   |   duration <= 1.512244: F (107.0/30.0)
|   |   |   duration > 1.512244: T (15.0/1.0)
|   |   e_range > -0.780483: T (1983.0/502.0)
```

**Fig. 1.** Decision tree C4.5. Simplified version with pruning confidence of 0.001 (default is 0.25). T in the branches is accented word and F is unaccented word.

**Table 5.** Most frequent consistencies and inconsistencies. Rows refer to the corpus used to train the classifier. Columns refer to the groups of labelers to contrast with. In the cells the information is A/B(C), being A the label set by the human transcriber, B the label assigned by the automatic classifier and C is the percentage of observations among all the inconsistencies (upper table) or among all the consistencies (bottom table).

Most common disagreements

| | Group 1 of labelers | Group 2 of labelers |
|---|---|---|
| ESMA-UPC | L*/noAccent (35.1)<br>H*/noAccent (12.4)<br>L+H*/noAccent (11.4)<br>L+>H*/noAccent (10.9) | L*/noAccent (35.3)<br>0/Accent (24.3)<br>L+H*/noAccent (13.2)<br>L+>H*/noAccent (11.8) |
| BURNC | L*/noAccent (36.0)<br>!H*/noAccent (10.3)<br>L+H*/noAccent (9.8)<br>H*/noAccent (9.8) | L*/noAccent (40.3)<br>0/Accent (19.4)<br>L+>H*/noAccent (10.9)<br>L+H*/noAccent (10.1) |

Most common agreements

| | Group 1 of labelers | Group 2 of labelers |
|---|---|---|
| ESMA-UPC | L+>H*/Accent (26.0)<br>0/noAccent (19.2)<br>L+H*/Accent (12.4)<br>L*/Accent (12.0) | 0/noAccent (35.0)<br>L+>H*/Accent (25.6)<br>L+H*/Accent (12.3)<br>L*/Accent (10.3) |
| BURNC | L+>H*/Accent (28.6)<br>0/noAccent (20.2)<br>L+H*/Accent (13.9)<br>H*/Accent (12.6) | 0/noAccent (37.6)<br>L+>H*/Accent (25.7)<br>L+H*/Accent (14.3)<br>L*/Accent (8.1) |

help to improve the inter-rater agreement or to resolve labeling inconsistencies. In this example, the automatic predictions are clearly closer to the second group of labelers with differences of 18.2 points average using the ESMA-UPC as the training corpus and 12.8 using the BURNC corpus, an objective indicator being to select a labeler or the labels from one of the two groups.

This mentioned objective criterion can be defended by using the resulting decision trees (see figure 1). The essential interpretation of the decision trees has been observed to be the same for both languages: the *accented* tag is assigned to combinations of features with a high variability of F0 and/or energy and/or long duration. The cross-lingual observed differences affects to the position of the features in the levels of the trees and to the threshold values used in the binary decisions. As the classification results are contrasted, a common inter classifier behavior is observed. Thus for example, table 5 shows that the most frequent disagreement is the same both for the ESMA-UPC trained classifier and the BURNC trained one: A high number of L* tonal accents are classified as *unaccented* which represents more than 35% of all the disagreements in both cases. Furthermore, the four most common disagreements, representing more than 80% of the total amount of disagreements with respect to the symbols assigned by the group two of labelers, are shared by both classifiers. Again, the four most common agreements representing more than the 80% of the agreements are also the same for both classifiers. This result evidences a similar behavior of the classifiers, and encourages for using cross-lingual labeling of prosodic event in combination with a posterior supervised revision of the results by human labelers in future works.

## 5 Conclusions and Future Work

A cross-lingual experiment on tonal accent identification has been presented. The two corpora used have been presented and the experimental strategy has been described. Results indicate that the automatic classifiers offer an objective criterion that permits close to 75% of the input units in cross lingual situations to be identified. The inter-corpus input features scale differences force the revision of the results by the manual labelers. Nevertheless, the predictions of the classifiers can be considered as an objective reference to speed-up the ToBI labeling of the target corpus.

Cross-lingual differences appear in the relevance of the input features and their distribution of values with negative impact in the labeling results. To overcome this difficulty is the challenge for future work with the use of speaker adaptation techniques, introduction of more representative input features or the inclusion of language dependent information in the normalization process.

## References

1. Aguilar, L., Bonafonte, A., Campillo, F., Escudero, D.: Determining Intonational Boundaries from the Acoustic Signal. In: Proceedings of Interspeech 2009, pp. 2447–2450 (2009)
2. Ananthakrishnan, S., Narayanan, S.: Automatic Prosodic Event Detection Using Acoustic, Lexical, and Syntactic Evidence. IEEE Transactions on Audio, Speech, and Language Processing 16(1), 216–228 (2008)
3. Bonafonte, A., Moreno, A.: Documentation of the upc-esma spanish database. Tech. rep., TALP Research Center, Universitat Politecnica de Catalunya, Barcelona, Spain (2008)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
5. Escudero, D., Aguilar, L.: Procedure for assessing the reliability of prosodic judgements using Sp-TOBI labeling system. In: Proceedings of Prosody 2010 (2010)
6. Gonzalez, C., Vivaracho, C., Escudero, D., Cardenoso, V.: On the Automatic ToBI Accent Type Identification from Data. In: Interspeech 2010 (2010)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1), 10–18 (2009)
8. Meteer, M., Schwartz, R.M., Weischedel, R.M.: Post: Using probabilities in language processing. In: IJCAI, pp. 960–965 (1991)
9. Ostendorf, M., Price, P., Shattuck, S.: The boston university radio news corpus. Tech. rep., Boston University (1995)

10. Rangarajan Sridhar, V., Bangalore, S., Narayanan, S.: Exploiting Acoustic and Syntactic Features for Automatic Prosody Labeling in a Maximum Entropy Framework. IEEE Transactions on Audio, Speech, and Language Processing 16(4), 797–811 (2008)
11. Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J.: ToBI: A standard for labelling English prosody. In: Proceedings of ICSLP-1992, pp. 867–870 (1992)
12. Syrdal, A.K., Hirshberg, J., McGory, J., Beckman, M.: Automatic ToBI prediction and alignment to speed manual labeling of prosody. Speech Communication (33), 135–151 (2001)
13. Vivaracho-Pascual, Simon-Hurtado, A.: Improving ann performance for imbalanced data sets by means of the ntil technique. In: IEEE International Joint Conference on Neural Networks (July 18-23, 2010)
14. Wightman, C., Ostendorf, M.: Automatic labeling of prosodic patterns. IEEE Transactions on Speech and Audio Processing 2(4), 469–481 (1994)
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (1999)

# Automatic Semantic Labeling of Medical Texts with Feature Structures

Agnieszka Mykowiecka and Małgorzata Marciniak

Institute of Computer Science, Polish Academy of Sciences,
J. K. Ordona 21, 01-237 Warsaw, Poland
{agn,mm}@ipipan.waw.pl

**Abstract.** This paper presents the results of testing two approaches in the automatic semantic labeling of medical data. For a chosen domain (diabetic patients' discharge records) a set of domain related concepts was identified. The annotated resource is the result of a rule based application, that relies on the results of two related rule based information extraction (IE) systems, post processed in a way that makes the label structures simpler, and the boundaries of annotations more precise. The second application is a machine learning (CRF) approach in which the results of the first application are used as training data. Both applications were evaluated by comparing to manually corrected documents.

## 1 Introduction

The aim of the work presented in this paper was to test the possibility of automatic semantic annotation of Polish hospital discharge records. Such an application would be very useful for automatic selection of data written in free text within patients' documents.

Most current efforts undertaken in the domain of recognizing various types of information included within natural language texts, are done using statistical and machine learning approaches. Such applications need specially prepared domain data for training and testing. Although there have already been many projects aimed at creating annotated text datasets, for our task there was no already established resource available. The problem of data availability is especially true for medical clinical texts. It is not true only for Polish, but for any language. While there are quite a lot of biomedical scientific literature available, language used in every day communication between physicians is much less represented (especially in an annotated form). None of the Polish corpora includes this type of texts. 6 corpora described in [1] contain only biomedical texts. Only recently, some resources containing clinical data began to be constructed. The best known new corpus containing English clinical texts is CLEF [10]. It contains 20,000 cancer patient records annotated with information about clinical relations, entities, and temporal information. MEDLEX [4] is a Swedish medical corpus annotated with terminology and named entities. European project MedIEQ resulted in creation of the AQUA system which labels web resources with MeSH terms [3]. This year, the Text REtrieval Conference (TREC) added the Medical Records Track aimed at exploring methods for searching unstructured information in patient medical records.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 49–56, 2011.

There are two main approaches to the task of creating new linguistic data – manual annotation, and manual correction of automatically assigned labels. The process of manual construction of the annotated corpus is usually long and laborious. As manual work is prone to errors there are established standards of constructing such resources. But even applying these standards does not eliminate all errors or assure the consistency of a resource. Automatic annotation is much faster and although it also does not guarantee complete correctness, the cost of correcting already labeled data is lower. In our work we decided to use and modify a resource that was prepared on the basis of the results from an already existing rule based IE system. It contains semantic annotations of 460 diabetic patients discharge records. The data consists of 465004 tokens, of which 55% are words, abbreviations and acronyms, while the rest are numbers and punctuation marks. In section 2 we present the type of semantic information that is included in this resource. The details of the process of corpus construction, the changes introduced for the purpose of this work, and the results of verification of the resource based on manual correction of 10% of the data are given in section 3. The annotated data was used as a training set for the elaboration of a CRF model of semantic labels assignment. In section 4, we describe the experiments preformed, and the results for different choices of sets of input features.

## 2 Semantic Information Identified

Every discharge record is divided into up to seven sections: *Introduction, Diagnosis, Examinations results, Treatment, Discharge record, Recommendations* and *Sign*. The types of information that were identified and annotated, were selected by anexpert. This list contains about 50 facts, some of which have an internal structure. The selected facts can be divided into the following groups:

- administrative information (dates of a visit, identification numbers),
- basic patient information: age, sex, weight,
- data about diabetes, e.g,: type, duration, basic biochemical test results,
- complications and other illnesses, which may be correlated with diabetes,
- diabetes treatment: insulin type and its doses, and other oral medication,
- details concerning diet,
- patient education and level of his/her cooperativeness.

The recognized facts are represented as values of features defined in a small domain related ontology [6]. According to the annotation schema accepted, a contiguous sequence of words can be assigned either a simple label consisting only of a name and a value, representing for example a weight, or a feature-value structure consisting of a list of feature-value pairs, representing for example a name and a dose of medication. For some structures, only the external boundaries of the annotated phrase are given, for others, the alignment of the internal attributes and parts of the word sequence are also shown. Sometimes, the same information can be recognized as a stand-alone attribute, whilst in other contexts, it is included in a larger structure. The internal format of the annotation is based on the TEI standard – every document is represented in the form of several XML files describing subsequent layers of annotation. Figure 1 presents examples of a semantic annotation in the form of feature-value structures. The first example shows annotation of the sequence of words with one simple attribute (D_CONTROLL) assigned. In this example the word *cukrzycy* ‘diabetes’ indicates only a context in which the other two words are recognized. In the second structure the same attribute is recognized within the structure representing reasons for a patient’s visit in a hospital. In this case the sequence of five words has the *reaso_l_str* label assigned while next three have a *reaso_l_str*|D_CONTROLL path assigned. The third example shows a diet structure which has two internal structures for calories and meals limits.

*<niedostateczne wyrównanie cukrzycy>*
(*uncontrolled diabetes*)

$$\left[\begin{array}{l} \text{D\_CONTROLL} \quad \text{uncontrolled\_t} \\ < \mathit{niedostateczne\ wyrownanie} > \end{array}\right]$$

*<przyjety do kliniki z powodu wysokich wartości glikemii >*
(*admitted to the hospital because of hyperglycemia* )

$$\left[\begin{array}{l} \text{reaso\_l\_str} \\ \left[\text{D\_CONTROLL hiperglikemia\_t}\right] \\ < \mathit{wysokich\ wartosci\ glikemii} > \end{array}\right]$$

*<Dieta cukrzycowa 1500 kcal 6 posiłków>*
(*Diabetic diet 1500 kcal 6 meals*)

$$\left[\begin{array}{l} \text{diet\_str} \\ \left[\text{DIET\_TYPE d\_diab\_t}\right] \\ < \mathit{Dieta\ cukrzycowa} > \\ \left[\begin{array}{l} \text{cal\_str} \\ \text{CAL\_MIN 1500} \end{array}\right] \\ < \mathit{1500kcal} > \\ \left[\begin{array}{l} \text{meals\_str} \\ \text{MEALS\_MIN 6} \end{array}\right] \\ < \mathit{6\ posilkow} > \end{array}\right]$$

**Fig. 1.** Examples of labels structure

## 3 Data Preparation

The annotated resource was obtained by using a corpus constructed from the results of the already existing rule based extraction system. The IE system was evaluated on a set of 100 documents [8]. Although the results were good (f-measure 0.979 for 4021 identified attributes), it turned out that their structure was not entirely appropriate for corpus construction. For IE, the main goal was to find out whether a particular piece of information is present in an analyzed text. The task of text annotation requires a more precise strategy, that of identifying the boundaries of text fragments which are to be annotated with a given label. To solve the problem, the idea of combining two extraction grammars was proposed. The main idea consists in recognizing precise text pieces in one grammar, and verifying their correctness by a more complex grammar which looks deeper into the context. It can be summed up as follows:

- The full extraction grammar rules are applied to the text. The rules recognize information with a high precision and recall, but if a complex structure (consisting of several attributes, see examples in fig. (1)) is assigned to a phrase, we don't know which part of the phrase represents a particular attribute.

- The simplified extraction grammar rules are applied to the text. The rules recognize small pieces of information, usually one attribute, only numerical values (e.g. ranges, dates) are still recognized together. The rules do not look into context, so for example the phrase *10 lat temu* '10 years ago' is always recognized by this grammar as information when diabetes was diagnosed, while it can refer to another ailment like *nadciśnienie zdiagnozowane 10 lat temu* 'hypertension diagnosed 10 years ago' or even other information *hospitalizowany 10 lat temu* 'hospitalized 10 years ago'.
- The results of both extraction grammars are cleaned up. Non-informative pieces of structures are removed from the results. For example, if in a recommended diet only one quantity of calories is given, it is assigned to the CAL_MIN attribute, while the CAL_MAX attribute is assigned a general *string* value that can be removed. Moreover, if an attribute has assigned a label indicating that its value is unified with the value of another structure, this label is replaced by the value itself.
- The results of both grammars are compared and only structures that are represented in both results are represented in the final corpus data. We take into account boundaries of phrases from both grammars. Broad phrases (from the full grammar) indicate the context in which narrow phrases (from the simplified grammar) are informative.

For the current experiment, further refinements to the annotation process were introduced. The changes mainly concern the problem of inconsistently recognized phrases. For example the abbreviation of a year (*r*) was sometimes recognized with, and sometimes without, a subsequent dot. In the case of creatinine level test results, only part of a complex unit was included in the phrase to which the information was attached. Such inconsistencies could cause problems in determining the principles for the manual verification of annotation results, and difficulties in evaluating an experiment's annotation. An additional post processing phase aimed at simplifying the labels' final structure was also introduced. At this stage, structures with only one possible internal element were flattened to this internal one attribute or structure.

In the corpus, 66165 simple attribute values are annotated, 43789 of them are elements of complex feature-value structures. Table 1 shows the numbers of occurrences of all types of structures. Numbers for simple attributes are the sums for their occurrences in isolation or inside complex structures.

Manual verification of 10% of the corpus (46 records, 46439 tokens) was done by 2 annotators, according to instructions prepared by the person developing the extraction grammars, [7]. Annotators should correct all labels together with their limits. Annotators had to point out not only information that was constructed according to the described rules, but also other ways of representing adequate information if it was understandable to human readers. Another important guideline for annotators recommended they ignore understandable spelling errors which caused the information to not be recognized by the grammars.

The results of manual correction were compared, and one coherent version was elaborated in the inter-annotators' negotiations. The final changes concerned 596 token

**Table 1.** Semantic labels occurrences

| structure/attribute | number of occurrences | nb of tokens labeled | nb of possible values | nb of different phrases types |
|---|---|---|---|---|
| administrative information | | | | |
| DOC_BEG | 460 | 920 | 1 | 2 |
| DOC_DAT | 390 | 3119 | (date) | 3 |
| id_str | 457 | 1833 | (number/symbol) | 3 |
| hospit_str | 436 | 7821 | – | 11 |
| H_FROM | 436 | 3053 | (date) | 3 |
| H_TO | 436 | 3056 | (date) | 2 |
| EPIKRYZA_BEG | 459 | 459 | 1 | 1 |
| recommendation_str | 445 | 2471 | – | 57 |
| RECOMMENDATION_BEG | 443 | 886 | 1 | 2 |
| basic patient data | | | | |
| id_pat_str | 443 | 1838 | – | 3 |
| id_pat_sex | 443 | 944 | (symbol) | 3 |
| ID_AGE | 903 | 2237 | (number) | 6 |
| W_IN_WORDS | 247 | 247 | 2 | 8 |
| WEIGHT | 390 | 1424 | (number) | 6 |
| BMI | 328 | 1101 | (number) | 7 |
| HEIGHT | 390 | 807 | (number) | 5 |
| basic diabetes data | | | | |
| D_TYPE | 738 | 1461 | 3 | 10 |
| D_CONTROLL | 867 | 1615 | 4 | 81 |
| ABSOLUT_DATA | 11 | 33 | (date) | 3 |
| FROM_IN_W | 146 | 146 | 2 | 9 |
| RELATIVE_DATA | 153 | 463 | (number/unit) | 22 |
| YEAR_OF_LIFE | 12 | 49 | (number) | 4 |
| HBA1C | 511 | 2519 | (number) | 12 |
| KWAS_D | 18 | 36 | 2 | 2 |
| ACET_D | 469 | 987 | 2 | 24 |
| KETO_D | 11 | 12 | 2 | 4 |
| creatinin_str | 430 | 1865 | (numbers) | 11 |
| microalbuminury_str | 80 | 413 | (numbers) | 16 |
| lipid_str | 259 | 6270 | – | 23 |
| LDL1 | 247 | 540 | (number) | 3 |
| reason | 279 | 2997 | – | 151 |
| complication and acc diseases | | | | |
| ACC_DISEASE | 491 | 491 | 1 | 3 |
| COMP | 1327 | 2654 | 16 | 122 |
| N_COMP | 191 | 1020 | 18 | 32 |
| AUTOIMM_DISEASE | 13 | 16 | 3 | 2 |
| therapy | | | | |
| insulin_treat_str | 4387 | 24896 | – | 321 |
| I_TYPE | 4910 | 7987 | 51 | 61 |
| dose_str | 4339 | 13330 | (range) | 16 |
| diet_str | 671 | 3608 | – | 93 |
| DIET_TYPE | 421 | 857 | 7 | 9 |
| cal_str | 421 | 1149 | (range) | 12 |
| meals_str | 388 | 890 | (range) | 9 |
| insulin_inf_treat | 31 | 220 | (range) | 14 |
| bolus_b_m | 7 | 40 | (range) | 7 |
| ORAL_TREAT | 1026 | 1177 | 36 | 41 |
| D_TREAT | 275 | 581 | 3 | 28 |
| I_THERAPY_BEG | 27 | 88 | 1 | 11 |
| THERAPY_MODIFF | 184 | 711 | 2 | 90 |
| DOSE_MODIFF | 65 | 195 | 2 | 30 |
| DIET_CORRECTION | 18 | 51 | 2 | 12 |
| DIET_OBSERVE | 7 | 22 | 2 | 2 |
| SELF_MONITORING | 13 | 36 | 2 | 8 |
| EDUCATION | 355 | 2914 | 1 | 189 |

labels (1.3%), 283 of them were changed in the same way by both annotators. Changes concerned mainly adding new labels (79). The kappa coefficient counted for non-empty labels on the words level was equal to 0.966. The results of the comparison of the automatically annotated corpus and the manually corrected version counted on 3309 structures (simple or complex) showed 0.961 accuracy, precision: 0.994, recall: 0.966, f-measure: 0.98. F-measure counted for 9057 non empty word–label pairs was equal to 0.98.

## 4 CRF Model of Label Assignment

The annotated resource can serve as a training set to prepare a machine learning application for automatic labeling. For the first experiments, among many existing techniques, the Conditional Random Field method (CRF, [11]) was chosen. It proved to be very good for the task of assigning many types of labels to transcribed speech [2] and was already successfully used to annotate Polish dialogues in the domain of public transport [5], [9].

The specificity of the CRF method allows to build the model using very many attributes describing particular input items. As these features, we used the forms, POS names, lemmas, morphological features and token types. They might concern the described element itself or its left or right context. We also tried to include values of attributes as part of labels, and to distinguish the first word attributed with a label from the subsequent ones. The CRF method assigns one label to every input token, and does not distinguish any internal label structure or attribute values. In case of structured labels, we preformed two types of linearization using only the most internal attributes or the entire paths.

In our experiments we used 368 records out of 460 as a training set, 46 as a test set and (the same as before) 46 documents as an evaluation set. Several combinations of up to 8 features per token were tested. The overall results for the best models are given in Table 3. The results of verification done on the automatically annotated records and on the same set of documents after their manual correction (marked in column 3) turned out to be very similar.

The best model which assigns the simplified (short) labels (number 4) uses a part of speech name, a lemma, two preceding and two following tokens. Adding morphological features (case or gender) lowered the results. The detailed results of model 4 validation

**Table 2.** Evaluation results for different CRF models

| | features | eval.set corr. | Acc. | Prec. | Recall | F-measure |
|---|---|---|---|---|---|---|
| 1 | paths, token type, context [-3, +2] | - | 0.8392 | 0.9525 | 0.5336 | 0.6805 |
| 2 | paths, token type, context [-3, +2] | + | 0.8392 | 0.9250 | 0.5507 | 0.6903 |
| 3 | paths, token types [-1,+1], context [-3,+2] | + | 0.9756 | 0.9739 | 0.8989 | 0.9349 |
| 4 | short labels, POS, lemma, context [-2, +2] | + | 0.9768 | 0.9748 | 0.9044 | 0.9383 |
| 5 | short labels, token types [-1,+1], context [-3,+2] | + | 0.9757 | 0.9735 | 0.8996 | 0.9351 |
| 6 | short labels, token types, [-1,+1], context [-2,+2], value | + | 0.9814 | 0.9397 | 0.9582 | 0.9489 |
| | 2005-2006 training set (170 doc.) | | | | | |
| 7 | short labels, POS, lemma, context[-2, +2] | + | 0.9774 | 09142 | 0.9671 | 0.9399 |

**Table 3.** Best and worst results of the labels assignment for the model 4

| Concept | F-measure | precision | recall | ref. file | hyp. file |
|---|---|---|---|---|---|
| recommendation_str | 1.00 | 1.00 | 1.00 | 160 | 160 |
| id_pat_sex | 1.00 | 1.00 | 1.00 | 96 | 96 |
| creatinin_str | 1.00 | 1.00 | 1.00 | 182 | 182 |
| W_IN_WORDS | 1.00 | 1.00 | 1.00 | 15 | 15 |
| ACC_DISEASE | 1.00 | 1.00 | 1.00 | 48 | 48 |
| I_TYPE | 0.99 | 0.99 | 1.00 | 797 | 804 |
| dose_str | 0.99 | 1.00 | 0.99 | 1364 | 1356 |
| ... | | | | | |
| THERAPY_MODIFF | 0.91 | 0.84 | 1.00 | 62 | 74 |
| cure_l_str | 0.90 | 0.86 | 0.93 | 214 | 232 |
| DOSE_MODIFF | 0.88 | 0.85 | 0.92 | 25 | 27 |
| RELATIVE_DATA | 0.88 | 0.83 | 0.94 | 52 | 59 |
| lipid_str | 0.75 | 0.73 | 0.77 | 596 | 624 |
| microalbuminury_str | 0.73 | 0.58 | 1.00 | 33 | 57 |
| reaso_l_str | 0.73 | 0.62 | 0.90 | 165 | 240 |
| LDL1 | 0.63 | 0.47 | 0.96 | 28 | 58 |
| I_THERAPY_BEG | -1.00 | -1.00 | -1.00 | 1 | 6 |

are shown in Table 3. For long labels (paths) the best model (3) was based on token types of two neighbors and a little longer context. Surprisingly, results for this model were nearly as good as for the model 4. In this case using a part of speech and a lemma in place of token types did not result in a reliable model. Model number 6 presents results achieved for adding attribute values (only those which were not numbers) to the word labels. For full path names we did not obtained any stable model assigning attribute values.

## 5 Conclusions

The best computed CRF model showed nearly 0.94 of f-measure. The number will probably be even better if we manually correct the entire training set, but obtaining much higher results for long labels assignment might be difficult because of the task specific annotation structure. The scope of text which is needed for the correct label recognition is frequently not limited to the neighbouring words. Two very similar phrases could be annotated differently, or only one of them assigned a label, depending on the larger context.

The achieved results show that even using partially incorrect data for training, we can obtain a labeling model good enough for recognizing many complex labels (the original feature structures can be relatively easily reconstructed from the CRF output). While statistical models seem to be easier to apply for new data, rule based systems can be easier to control for recognizing long complex statements. They have also proved to be a good choice at the step of preparing an initial annotated resource. However, the quality of the CRF output proved this method to be useful for practical applications in the chosen domain, as it turned out to be dependent only on very simple features (token types) which can be easily computed for any new data set.

**Acknowledgements.** This research is supported by the POIG.01.01.02-14-013/09 project co-financed by EU under the European Regional Development Fund.

## References

1. Cohen, K.B., Fox, L., Ogren, P.V., Hunter, L.: Corpus design for biomedical natural language processing. In: ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, Detroit, pp. 38–45 (2005)
2. Hahn, S., Lehnen, P., Ney, H.: System combination for spoken language understanding. In: INTERSPEECH 2009. ISCA, Brisbane (2008)
3. Karkaletis, V., et al.: Automating accreditation of medical web content. In: Proceeding of the 18th European Conference on Artificial Intelligence (2008)
4. Kokkinakis, D.: A Semantically Annotated Swedish Medical Corpus. In: Proceedings of the LREC Conference, pp. 32–38 (2008)
5. Lehnen, P., Hahn, S., Ney, H., Mykowiecka, A.: Large scale Polish SLU. In: INTERSPEECH 2009. ISCA, Brighton (2009)
6. Mykowiecka, A., Marciniak, M.: Domain model for medical information extraction – the LightMedOnt ontology. In: Marciniak, M., Mykowiecka, A. (eds.) Bolc Festschrift. LNCS, vol. 5070, pp. 333–357. Springer, Heidelberg (2009)
7. Mykowiecka, A., Marciniak, M.: Some remarks on automatic semantic annotation of a medical corpus. In: Proc. of Third Louhi Workshop on Health Documentation Text Mining and Information Analysis at AIME (2011)
8. Mykowiecka, A., Marciniak, M., Kupść, A.: Rule-based information extraction from patient's clinical data. Journal of Biomedical Informatics 42, 923–936 (2009)
9. Mykowiecka, A., Waszczuk, J.: Semantic annotation of city transportation information dialogues using CRF method. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS (LNAI), vol. 5729, pp. 411–418. Springer, Heidelberg (2009)
10. Roberts, A., et al.: Building a semantically annotated corpus of clinical texts. Journal of Biomedical Informatics 42(5), 950–966 (2009)
11. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning. MIT Press, Cambridge (2007)

# Automatic Switchboard Operator

Tomáš Valenta and Luboš Šmídl

Department of Cybernetics, University of West Bohemia
Univerzitní 8, Pilsen, 306 14, Czech Republic
{valentat,smidl}@kky.zcu.cz

**Abstract.** This paper describes the Automatic Switchboard Operator system and experiences and improvements based on data collected while full operation of the application. Automatic Switchboard Operator is a voice dialogue application whose purpose is to answer phone calls and transfer the calls to the requested person. Called person is recognized according to a speech grammar which has key effect on successfulness of the system as a whole. After several months of full operation of the application, the speech grammar was made more robust in order to accept more utterances and filter out substantial information, i.e. a filler model was introduced.

**Keywords:** voice dialogue systems, automatic speech recognition, text-to-speech, VoiceXML.

## 1 Introduction

Automatic Switchboard Operator is a voice application whose purpose is to answer phone calls and transfer the calls to requested persons. The caller makes input by voice and the system informs him or her by voice as well.

The application was originally designed and developed for university environment to cover its entire phone book (about 2,000 persons). But its design allows deployment to any company including those using e.g. ranks (police, fire-fighters etc.).

In section 2 we describe a voice dialogue system and the dialogue from the user's point of view. Section 3 describes the key algorithms the system is built upon, especially rule-based utterances generation which the speech grammar is build upon. We also introduce core speech grammar for person input. In section 4 we discuss previous state of the project before grammar adjustments and put some practical experiences gained during the full operation of the system and their impact on speech grammar adjustments. Our observations are illustrated with some numbers. Finally in section 5 we summarize the results and close this paper.

## 2 Dialogue System and the Dialogue

The Automatic Switchboard Operator dialogue system uses VoiceXML interpreter, TTS unit, ASR unit and telephone back-end which supports both ISDN and SIP telephony, see figure 1. As soon as the call is answered, the interpreter fetches a document from

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 57–63, 2011.

a document server which uses a scripting engine to generate the contents and starts processing it.

VoiceXML interpreter, a core part performing the dialogue logic, and VoiceXML language is a powerful tool that can be used to design very complex dialogue systems. [4]

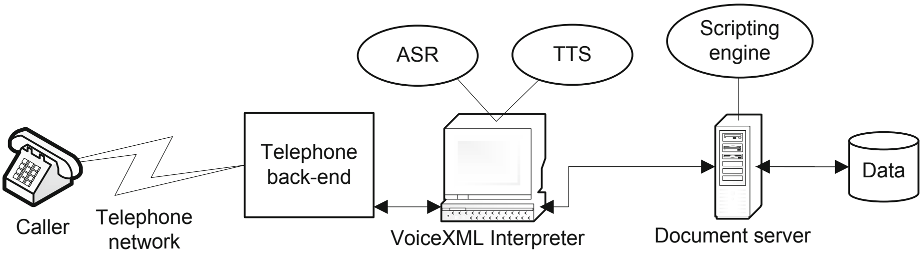


**Fig. 1.** Scheme of a dialogue system

As a document server, any web server with PHP scripting engine can be used. As a data source, MySQL database server is currently used. But the database module can be easily adapted to connect to any appropriate data source.

Automatic Switchboard Operator has two more parts not drawn in figure 1. They are *data importer* and *grammar compiler*. The purpose of data importer is to import phone book data from any data source.

The grammar compiler is a must for large phone books. In order to recognize (match utterances) against a speech grammar, the grammar must be compiled in advance. In case of few items, the compilation is fast, almost instant, but for large grammars, it can take tens of minutes. [1] The grammar is a context-free grammar, see below.

### 2.1 Scheme of the Dialogue

In the figure 2, the scheme of the dialogue is presented. It follows guidelines and techniques described in [5]. First, the caller is told that he/she is speaking to the Automatic Switchboard Operator. Then he/she is asked to enter the person to call. Surname or function must be entered, while other information such as first name, degrees or department the person is working at, may be used to narrow the search results. Alternatively the input can be performed by the telephone keyboard. It is the speech grammar for this stage that will be discussed in this paper.

## 3 Key Algorithms

The most important part of the Automatic Switchboard Operator application is to transform phone book data that are aimed to be presented visually into data that can be used aurally. Basically it means rewriting words with irregular pronunciation (e. g. abbreviations) phonetically. It is also useful to generate more aural alternatives to a word so that a user has more freedom what to say and the system will understand him/her.

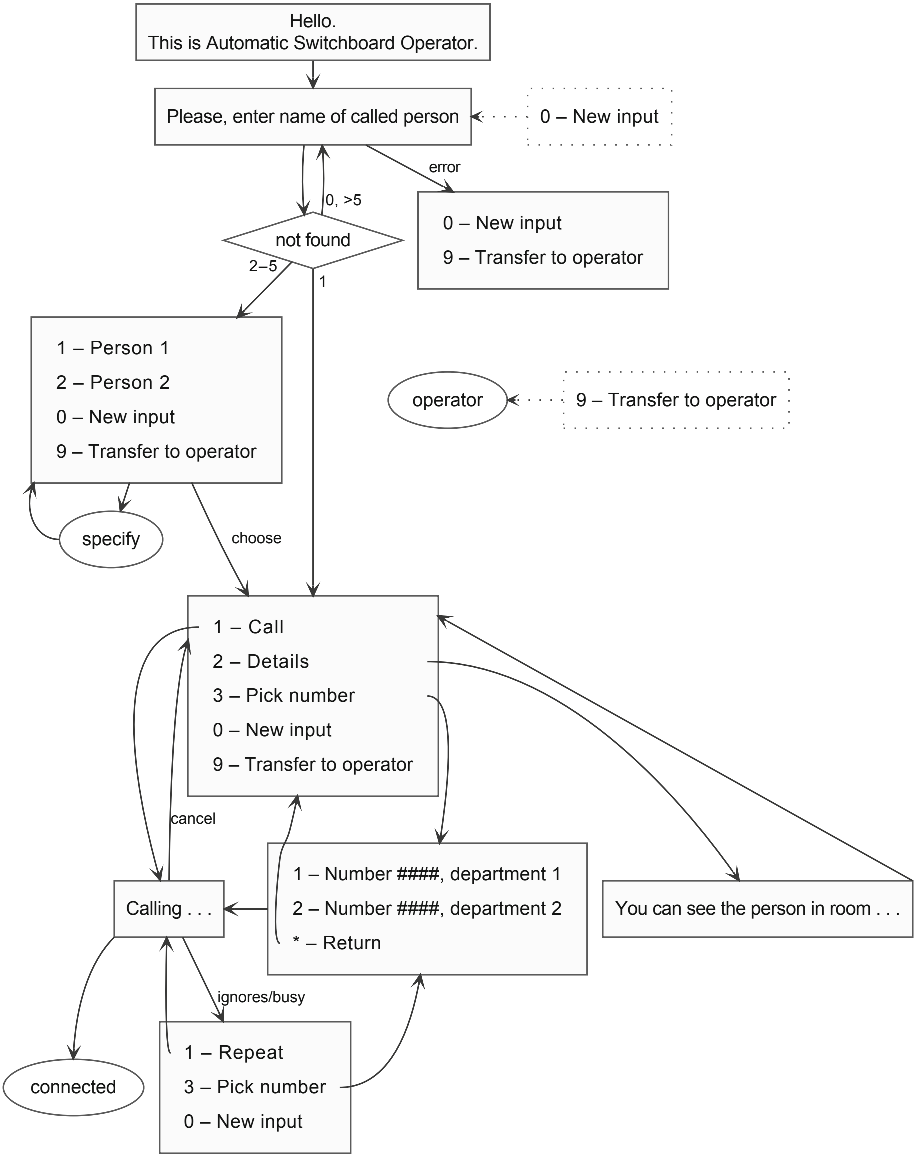


**Fig. 2.** Scheme of Automatic Switchboard Operator's dialogue

It involves text preprocessing, rule-based generation of synthesized phrases, rule-based generation of recognized utterances and DTMF sequences generation. The generated aural data is then used to build a grammar and to synthesize full person's title (including degrees, salutation etc.).

### 3.1 Text Preprocessing

Its goal is to remove all elements that have no effect on pronunciation. They are letter case (the text is converted to lower case), punctuation such as brackets, hyphens, commas, quotes etc. and unnecessary whitespace. [2] For example:

```
Ing.␣␣John␣Black-Smith,␣chief␣accountant␣
                    → ing.␣john␣black␣smith␣chief␣accountant
```

### 3.2 Rule-Based Generation of Utterances

This algorithm is used to generate phrases that are synthesized to inform the user that particular person was found or to generate the recognized utterances. The utterance consists of various fields, first name, surname, degrees, department and function. In the first step, each field is transcribed separately, which usually gives more transcriptions with certain amount of points. The points are given during the text replacement and evaluate best transcription for synthesis, whereas other utterances are recognized as well.

Rules example:

| | | | |
|---|---|---|---|
| rndr. | → | doctor of natural sciences | (MF) |
| doctor | → | he doctor | (M) |
| doctor | → | she doctor | (F) |

Rules are applied in each possible order and are gender-dependent, which is a must for Czech language.

For example with rules $James \rightarrow James$ (•), $James \rightarrow Jimmy$ ( ) [simplified], a person with first name James will be called James (synthesized by TTS), but Jimmy will also be recognized.

### 3.3 The Speech Grammar and Its Complexity

The speech grammar (a context-free grammar) can be represented as a directed graph shown in figure 3. It shows that surname and/or function of a person must be entered, other fields are voluntary and help narrow the search results, i.e. disambiguate persons.

It is also used to parse the utterance semantically, i.e. each word uttered is tagged with its meaning (J, P, T, D, F; see figure 3) so that database queries can be performed.

A formula for number of accepted utterances for a person can be derived from the graph (assuming the user will not say the same thing twice). For example a person with one salutation (Mr.), three degrees that can be transcribed in five ways, one first name, one surname, two departments and two functions gives us 18,602 utterances accepted by the grammar.

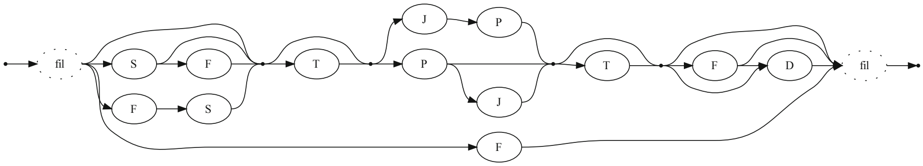


**Fig. 3.** Speech grammar for person input. S. . . salutation, F. . . function, T. . . degree, D. . . department, J. . . first name, P. . . surname, fil. . . filler model (dotted)

For a phone book containing 2000 entries (persons) it gives roughly 20 million utterances accepted by the whole grammar active during the input of a name of a person.

The chosen method (using a context-free grammar with labelling) was the most suitable one for such application, because there is not enough data for training a statistical parser or to build a language model covering all the names which can change throughout the time as the phone book changes. [3]

While nonspeech events, such as coughing, laughing etc., are filtered out by speech/ nonspeech detector, a part of ASR unit [1], speech (word) filler model has to be implemented by the grammar. The filler model will be discussed in the next section.

## 4 Experiences and Grammar Adjustments

According to the data collected and annotated from May 2008 till December 2009, during the full operation of the application, many utterances were rejected by the grammar because of a missing filler model. That is why the filler model was introduced to the grammar.

Many callers started the call with greetings, polite requests, thanks and even swear-words. A filler model was introduced to the grammar to filter them out and return a clean parsed person search request.

The accepted (but not parsed and returned) utterances mainly consist of “hello”, “I would like. . . ”, “Can you connect me to. . . ”, “please”, “thank you”, “well. . . ” etc. If there is a correct query besides these “politenesses” (well, sometimes there are already mentioned swear-words which are accepted as well), it is matched and returned for the database query.

Before introducing the filler model, the speech grammar (finite state automaton) had 50,584 states and 148,720 edges with 1,872 final states. After introducing the filler model, the problem became even more complex: the number of states is now 76,177 and number of edges 448,100 with 13,406 final states.

### 4.1 Statistics Collected during the Operation

Within the 20 months of operation, 11,043 calls were answered by the Automatic Switchboard Operator. From those, 8,920 (81.78 %) were out-of-grammar utterances, see below. The rest (2,123) were utterances accepted by the grammar.

The accepted utterances were compared to the data transcribed by annotators with 88.45 % word correctness. These errors involve misunderstood words with very similar pronunciations (e.g. Drábová and Drábková in Czech).

### 4.2 Out-of-Grammar Utterances

Utterances rejected by the grammar can be divided into four groups:

- Nonsense – utterances without an adequate phone book query, i.e. just swear-words, wrong dialled numbers etc.
- Silence, noise.
- Commands from another dialogue phase – in the figure 2 you can see that the dialogue consists of several phases. Some users were confused and uttered commands for different phase of the dialogue instead of entering a person query, for example "New input".
- Foreign language – only Czech is understood by the system so far.

For people that cannot perform their input by Czech voice, there are alternative ways how to get transferred to the desired destination. First they can get connected with human operator.

For those who cannot enter the query by voice because of noises in the background, they can perform the input by phone keyboard in T9-style. For example if they want to enter "Smith", they enter "76484#", the hash sign means end of keyboard input.

## 5 Summary

The Automatic Switchboard Operator is a mature project which has gone through several stages of its life cycle. They are idea, design, development, testing, operation, evaluation and fine-tuning. It is available 24 hours a day 7 day a week, thanks to the data importer always works with recent phone book and is proven to work well.

It provides its service by voice, the most natural way for people to communicate. If the voice communication is not possible, an alternative way of input is provided: the touch tones (phone keyboard). It also has a fall back to a human operator.

It is closely tied with Czech language and its processing, but for companies having mostly Czech employees and expecting mainly Czech callers there is an option to transfer foreign callers to operators speaking their language: The well-known prompt "For English, press star."

Installing application in a new environment involves nothing more than installation of required components (web and database servers), brief configuration, changing some texts (e.g. company name) and configuring the data importer to properly transform company's phone book into the expected data structure (usually just single SQL query).

The voice recognition module works with correctness 88.45 %, which was tested on data collected during 20 months of full operation. This shows that it is successful even with such complicated words like surnames.

Introducing the application into a company can be a very good investment.

**Acknowledgements.** The work has been supported by the Ministry of Education of the Czech Republic under project No. 1M0567 and by the grant of the University of West Bohemia, project No. SGS-2010-054.

## References

1. Müller, L., Psutka, J., Šmídl, L.: Design of speech recognition engine. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2000. LNCS (LNAI), vol. 1902, pp. 259–264. Springer, Heidelberg (2000)
2. 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)
3. Jurčíček, F., Švec, J., Müller, L.: Extension of HVS semantic parser by allowing left-right branching. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2008, pp. 4993–4996. IEEE, Las Vegas (2008)
4. Bečvář, P., Šmídl, L., Psutka, J., Pěchouček, M.: An Intelligent Telephony Interface of Multiagent Decision Support Systems. IEEE Transactions on Systems, Man, and Cybernetics 37(4), 553–560 (2007)
5. Balentine, B., Morgan, D.P.: How to Build a Speech Recognition Application, A Style Guide for Telephony Dialogues (1999) ISBN: 0-9671278-1-5

# Automatic Topic Identification for Large Scale Language Modeling Data Filtering

Lucie Skorkovská, Pavel Ircing, Aleš Pražák, and Jan Lehečka

University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics
Univerzitní 8, 306 14 Plzeň, Czech Republic
{lskorkov,ircing,aprazak,jlehecka}@kky.zcu.cz

**Abstract.** The paper presents a module for topic identification that is embedded into a complex system for acquisition and storing large volumes of text data from the Web. The module processes each of the acquired data items and assigns keywords to them from a defined topic hierarchy that was developed for this purposes and is also described in the paper. The quality of the topic identification is evaluated in two ways - using classic precision-recall measures and also indirectly, by measuring the ASR performance of the topic-specific language models that are built using the automatically filtered data.

**Keywords:** topic identification, language modeling, automatic speech recognition.

## 1 Introduction

Statistical language models (LM) that constitute the state-of-the-art language modeling technique in many areas of natural language processing (automatic speech recognition, machine translation, etc.) require an extensive amount of training data in order to ensure the robust estimation of their parameters. It might seem that the problem of data availability has already disappeared as the quantity of electronic texts available on-line nowadays exceeds every conceivable limit. However, when we want to use those data for language modeling, there are still several important problems that have to be solved. We must tackle the technical issues related to the actual download of the on-line content, the algorithms for stripping of the HTML (or other) markup, methods for text tokenization and normalization and, last but not least, also the detection of possible duplicate documents. The system that deals with the mentioned tasks is introduced in [8].

Once we have the "cleaned" data available, it is still not practical to use them for language modeling right away. First, the data are typically huge to the extent that it complicates the actual language model construction. Even more importantly, there is the evidence that the data quantity by itself might not be sufficient for good language model performance and what is more important is the right scope of the LM training texts. When the topic of the LM target domain is really specific, it happens that the "in-domain" language model estimated on a moderate-sized corpus vastly outperforms the model built using the data that are one or two orders of magnitude bigger but constitute just a general corpus [5].

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 64–71, 2011.

Thus, when we download and store texts that are meant for future LM training, the information about the document topic is extremely valuable but, at the same time, often not available from the data source. This paper therefore introduces a method for automatic identification of the document topic and presents two different evaluation scenarios for determining the method efficiency.

## 2 Topic Identification

As mentioned before, the main purpose of our topic identification module is to filter the huge amount of data according to their topics for the future use as the LM training data. We decided that more than one topic should be assigned to each article in our database and that the topics should form some sort of hierarchical system - a topic tree.

The topic identification module is a part of the system described in [8], each newly downloaded article is preprocessed by that system's algorithms before automatic topic identification starts. One of the problems that we have to solve is how many topics we should assign to each article. For the current version of the algorithm we have experimentally chosen to assign 3 topics to each article.

### 2.1 Topic (Keyword) Tree

When we started to design our topic identification module, we searched for some kind of an existing topic hierarchy, but we found out that there is no such suitable hierarchical system. Consequently, we have build our own topic hierarchy in the form of topic tree, based on our expert findings in topic and keyword distribution in the articles on the favourite news servers like *ČeskéNoviny.cz* or *iDnes.cz*.

At present the topic tree has 32 main topic categories like `health`, `culture` or `sport`, each of this main category has its subcategories with the "smallest" topics represented as leaves of this tree. An example of a branch representing the topic category `justice & courts` from the topic tree can be seen on figure 1.

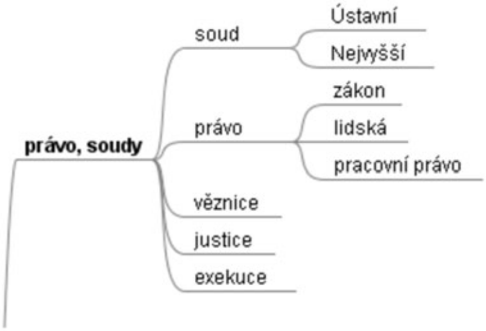


**Fig. 1.** Branch of the topic tree representing the topic `justice & courts`

In the current system, we use the topic tree with about 450 topics and topic categories, which correspond to the keywords assigned to the articles on the mentioned news servers. The articles with these "originally" assigned topics are used as training text for identification algorithms.

### 2.2 Identification Algorithms

Two methods for automatic topic identification was implemented so far, a classification based on TF-IDF[1] vector space model and a language modeling based classification. These methods were selected due to the good results in our information retrieval experiments [2], since we had no experience with the topic identification task so far.

**Language Modeling Based Classification.** The language modeling based approach chosen for the first experiments is similar to the Naive Bayes classifier [3], where the probability $P(T|A)$ of an article $A$ belonging to a class (topic in our case) $T$ is computed as

$$P(T|A) \propto P(T) \prod_{t \in A} P(t|T) \tag{1}$$

where $P(T)$ is the prior probability of a topic $T$ and $P(t|T)$ is a conditional probability of a term $t$ given the topic $T$. This probability can be estimated by the maximum likelihood estimate simply as the relative frequency of the term $t$ in the training articles belonging to the topic $T$:

$$\hat{P}(t|T) = \frac{tf_{t,T}}{N_T} \tag{2}$$

where $tf_{t,T}$ is the frequency of the term $t$ in $T$ and $N_T$ is the total number of tokens in articles of the topic $T$.

The goal of this language modeling based approach is to find the most likely or the maximum a posteriori topic (or topics) $T_{map}$ of an article $A$:

$$T_{map} = \arg\max_T \hat{P}(T|A) = \arg\max_T \hat{P}(T) \prod_{t \in A} \hat{P}(t|T) . \tag{3}$$

The prior probability of the topic $\hat{P}(T)$ was implemented as the relative frequency of the articles belonging to the topic in the training set, but we found out that it has no effect on the identification results.

**Vector Space Model Classification.** The second tested algorithm is the TF-IDF vector space model based classification. For each term $t$ in the topic $T$ the term frequency $tf_{t,T}$ and inverse document frequency is computed:

$$idf_t = \log \frac{N}{N_t} \tag{4}$$

[1] Term Frequency - Inverse Document Frequency.

where $N$ is the total number of topics and $N_t$ is the number of topics containing the term $t$. The similarity of an article $A$ and a topic $T$ is then computed as:

$$sim(A,T) = \sum_{t \in A} tf_{t,T} \cdot idf_t \ . \tag{5}$$

The topics with the highest similarity are then assigned to the tested article.

### 2.3 Evaluation

For the evaluation of the chosen topic identification methods a smaller collection of articles from the news server *ČeskéNoviny.cz* was separated. This collection contains 158 000 articles, 140 000 of these articles were used as topic training data, remaining 18 000 is available for evaluation testing. The articles from *ČeskéNoviny.cz* have included the originally assigned keywords from their authors (in average 3.5 keywords for one article), which were used as the training and reference topics.

Two types of evaluation were performed on the test collection. The first one is more from the point of view of information retrieval (IR), where each newly downloaded article is considered as a query in IR and precision ($P$), recall ($R$) and $F_1$-measure is computed for the answer topic set:

$$P = \frac{T_C}{T_A}, \qquad R = \frac{T_C}{T_R}, \qquad F_1 = 2\frac{P \cdot R}{P + R} \tag{6}$$

where $T_A$ is the number of topics assigned to the article, $T_C$ is the number of correctly assigned topics and $T_R$ is the number of relevant reference topics. An average of these measures is then computed across a set of testing articles.

The second type of evaluation is from the point of view of a topic classifier, where $P$, $R$ and $F_1$ is computed for each topic separately. Two ways of computing the average measures can be applied in this case, *microaveraging* (topics count proportionally to the size of the topic article set):

$$P_{micro} = \frac{\sum_T T_C}{\sum_T T_A}, \qquad R_{micro} = \frac{\sum_T T_C}{\sum_T T_R} \tag{7}$$

and *macroaveraging* (all topics count the same):

$$P_{macro} = \frac{\sum_T P_T}{|T|}, \qquad R_{macro} = \frac{\sum_T R_T}{|T|} \tag{8}$$

In this case $T_A$ refers to the number of articles assigned to a topic, $T_C$ is the number of articles correctly assigned to the topic i.e. the "true positives", $T_R$ is the true number of articles with the topic and $|T|$ is the total number of topics. The *macroaverage* measures are more important in our case, because we want our classifier to perform well on infrequent topics, too.

First, we wanted to find out the best number of topics to assign to each article. The relation between the number of topics and $P$, $R$ and $F_1$ measures from the IR point of view is shown on figure 2, it can be seen that best results are obtained for 3 assigned

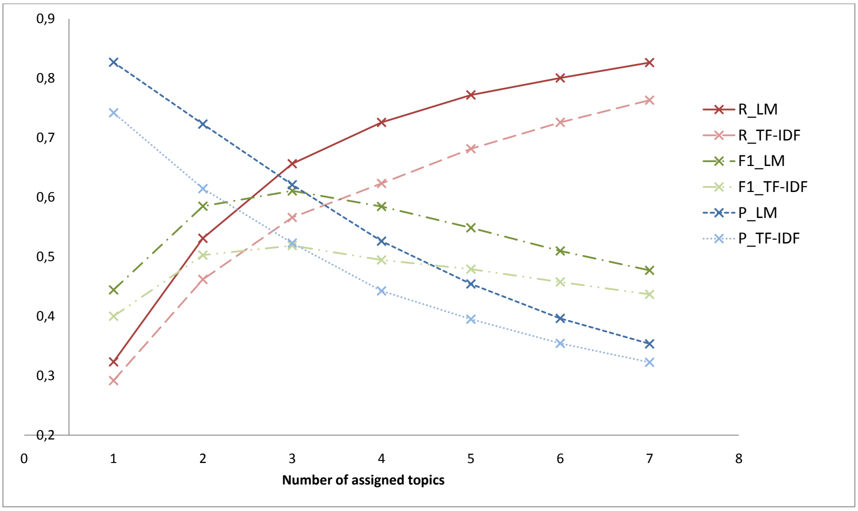


**Fig. 2.** Dependency of P, R and F1 on the number of assigned topics

**Table 1.** Average $P$, $R$ and $F_1$ measures of topic identification results for 15,000 set of articles

| | IR point of view | | | microaveraging | | | macroaveraging | | |
|---|---|---|---|---|---|---|---|---|---|
| **classification method** | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| **language modeling** | 0.594 | 0.626 | 0.583 | 0.597 | 0.570 | 0.583 | 0.624 | 0.442 | 0.517 |
| **vector space model** | 0.495 | 0.523 | 0.486 | 0.496 | 0.475 | 0.485 | 0.496 | 0.273 | 0.352 |

topics. $P$, $R$ and $F_1$ measures obtained for the test set of 15,000 articles and 3 assigned topics are shown in table 1.

The language modeling approach seems to achieve better results than vector space modeling, especially for topics with the small article set, which can be seen from the *macroaverage* $R$ and $F_1$ measures.

It may seem that the results are not so good, but it must be taken into consideration that we have a very large set of topics that are in many cases not well distinguished. Also the articles in the test collection are taken as they were on the news server, the original reference topics was not revised in any way, so in many cases the topic we assign to the article is also "correct", but it is not included in the reference set of topics. For example, the article about the achievements of the hockey representation has only `hockey` in reference topics, but our topic identification module assigned the topics `hockey, representation`, which is correct as well.

## 3 Language Modeling and ASR Experiments

The main motivation for the development of an automatic topic identification method introduced in the previous chapter was that we wanted to be able to effectively retrieve large amounts of domain-specific data for language model training. In this chapter, we will therefore present several experiments with language models estimated on the text corpora that were filtered from the large database of newspaper articles using various selection criteria. Since the ultimate measure of the language model quality is the performance of the system where the LM is employed (in this case the ASR decoder), we will also describe the speech recognition system that we have used and report relevant Word-Error-Rates (WER).

All language models perplexities (PPL) and WER were evaluated on test set consisting of speech obtained during the testing phase of the automatic closed-captioning system that employs the so-called “shadow-speaker” approach [4]. It means that the potentially noisy and/or overlapping broadcast speech is respoken with a trained speaker in controlled acoustic conditions in order to ensure higher recognition accuracy. The evaluation set contains recordings from just a single female speaker. The total length of the test set audio is 98 minutes. Since the speaker, whose utterances are in the test set, recorded in fact over 25 hours of data in total, we were able to tailor the acoustic models of the ASR system to this particular speaker (see [7] and [9] for details). This gives us a very high quality acoustic model and consequently, we can safely assume that any ASR performance gain from the improved language model would be even more prominent in the case when the acoustic model is less effective. All the language models described in the following paragraphs are trigram LMs estimated using the SRI Language Modeling Toolkit (SRILM) [6] employing the default Good-Turing discounting method. The resulting models always contain all the lexicon word bigrams that are found in the training data; the trigrams must occur at least twice to be included in the model.

The test set consists of samples of the dialogues that took place during the political talk show (“Otázky Václava Moravce”) broadcast by the Czech Television on July 18th, 2010. This particular show discussed mainly the newly appointed Czech government, state budget and also the health care issues. The appropriate keywords from the first-tier of the tree would then be `politics & diplomacy`, `economy` and `health`.[2] The first three lines of the Table 2 thus describe the language models that were trained using the articles published between January 1st, 2009 and July 17th, 2010 and are labeled with any keyword that comes from the subtree with the headword `politics & diplomacy`, `politics & diplomacy` and `economy`, and `politics & diplomacy`, `economy` and `health`. The results for these topic-specific LMs are compared with the models that are trained from all the articles that were published in the defined period just prior the broadcast day (lines 4 to 6). Such criterion is also a powerful filter of the retrieved data topics as the “hot” issues tend to be discussed across different mass media at the same time. The length of the periods was chosen in order to obtain roughly the same amount of selected data for both topic-defined and time-defined selections.

It can be seen from the results that topic-defined language models moderately outperform the time-defined ones in term of WER (8 to 12% relative improvement). It should be noted, however, that `politics & diplomacy` is the second most frequent of all first-tier topics and constitutes over 13% of the articles (there are 32 topic in the

[2] Note that assuming to know the topics before the actual broadcasting is not unrealistic - the main themes of each debate are published on the broadcaster website beforehand.

**Table 2.** Properties of language models trained on different data selections

| | Selection ID | # tokens | Lex. size | LM size [MB] | OOV [%] | PPL | WER [%] |
|---|---|---|---|---|---|---|---|
| 1 | politics & dipl. | 42M | 281k | 348 | 1.19 | 777 | 4.56 |
| 2 | pol.+econ. | 51M | 302k | 426 | 1.19 | 718 | 4.44 |
| 3 | pol.+econ.+heal. | 62M | 358k | 512 | 1.09 | 716 | 4.42 |
| 4 | 4-months | 37M | 307k | 313 | 1.31 | 1,167 | 5.23 |
| 5 | 5-months | 47M | 332k | 382 | 1.20 | 1,087 | 5.05 |
| 6 | 7-months | 67M | 378k | 551 | 1.17 | 983 | 4.80 |
| 7 | 7-months P.E.H. only | 28M | 279k | 268 | 1.37 | 850 | 4.97 |
| 8 | sport | 48M | 190k | 262 | 4.53 | 4,493 | 8.80 |

first level of the tree in total). Only the somehow fuzzy topic `relax` is slightly more frequent and thus the rather topicaly coherent `politics` articles probably dominate even the general language model.

In order to make the direct comparison between general and topic-specific corpus, we have further applied the keyword filter from line 3 to the selection from line 6. As can be seen from line 7, a 3.5% relative increase of WER was observed, however, the topic-filtered model size is only about 50% of the general one. Such finding is important for the potential future deployment of limited-resource ASR systems. Finally, to show that we really cannot just use any large-enough text data to train a good language model, we have performed a somehow extreme experiment by taking the articles labeled with the all the "sport" keywords. As you can see from line 8, the WER increased by almost a 100%.

## 4 Conclusions and Future Work

Both the evaluation of the topic identification accuracy itself and the indirect evaluation of the WER of the resulting topic-specific language models suggest that the topic identification algorithms presented in this paper work reasonably well. However, there is still some room for improvement. First, some topics are clearly ill-defined, especially the ones concerning geography - there are often either too broad (e.g. `USA`) or too narrow (e.g. `Vítkov` - small town connected with one xenophobic cause). Some other topics make a good sense intuitively, but are extremely hard for the automatic system to distinguish between as they use virtually identical phraseology (e.g. men's and women's tennis competitions `Davis Cup` and `Fed Cup`, respectively).

Second, we would like to implement some improvements for the presented topic identification methods like the k-NN classifier for the vector space model or the use of topic bigram LMs for the language modeling based classification and test the effects of these improvements on the topic identification results. More sophisticated methods like Support Vector Machines for text classification [1] could also be explored.

Finally, one of the most challenging improvements that we would like to include in the future version of the topic identification module is the automatic determination of the number of topics that will be assigned to each article. The number of the assigned topics should not be predefined, but it should be somehow related to the topic identification similarity score.

**Acknowledgments.** This research was supported by the Technology Agency of the Czech Republic, project No. TA01011264 and by the grant of the University of West Bohemia, project No. SGS-2010-054.

## References

1. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
2. Kanis, J., Skorkovská, L.: Comparison of different lemmatization approaches through the means of information retrieval performance. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 93–100. Springer, Heidelberg (2010)
3. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
4. Pražák, A., Loose, Z., Psutka, J., Radová, V., Müller, L.: Four-phase re-speaker training system. In: Proceedings of SIGMAP 2011, Seville (2011)
5. 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: Proceedings of Eurospeech 2003, Geneva, pp. 1821–1824 (2003)
6. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: Proceedings of ICSLP 2002, Denver, pp. 901–904 (2002)
7. Vaněk, J., Psutka, J.: Gender-dependent acoustic models fusion developed for automatic subtitling of parliament meetings broadcasted by the Czech TV. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 431–438. Springer, Heidelberg (2010)
8. Švec, J., Hoidekr, J., Soutner, D., Vavruška, J.: Web text data mining for building large scale language modelling corpus. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS(LNAI), vol. 6836, pp. 356–363. Springer, Heidelberg (2011)
9. Zajíc, Z., Machlica, L., Müller, L.: Robust statistic estimates for adaptation in the task of speech recognition. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 464–471. Springer, Heidelberg (2010)

# Automatic Translation Error Analysis

Mark Fishel[1], Ondřej Bojar[2], Daniel Zeman[2], and Jan Berka[2]

[1] Department of Computer Science
University of Tartu, Estonia
fishel@ut.ee

[2] Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University in Prague, Czechia
{bojar,zeman}@ufal.mff.cuni.cz
berka.jan@gmail.com

**Abstract.** We propose a method of automatic identification of various error types in machine translation output. The approach is mostly based on monolingual word alignment of the hypothesis and the reference translation. In addition to common lexical errors misplaced words are also detected. A comparison to manually classified MT errors is presented. Our error classification is inspired by that of Vilar (2006; [17]), although distinguishing some of their categories is beyond the reach of the current version of our system.

## 1 Introduction

Most efforts on machine translation evaluation so far concentrated on producing a single score – be it manual evaluation (HTER, fluency/adequacy, rank [5] or quiz-based evaluation [1]) or automatic metrics (WER, BLEU, NIST, METEOR, TER, SemPOS, LRscore, etc.). Such evaluation techniques are convenient for comparison of two versions of a system or of competing systems but they do not provide enough detail to steer further development of the system. Admittedly, some rough indication can be obtained from detailed outputs of such metrics, e.g. the unigram vs. full BLEU score reflect more of accuracy and fluency, respectively.

[17] proposed a simple classification of error types in MT output for manual marking of errors. [4] used a variant of this classification on the WMT09 dataset and compared the manually flagged errors to the post-edits of the same dataset as carried out during the shared task [6]. Both manual error flags as well as manual edits reveal similar differences between the systems, e.g. which one drops content words most, which one fails to produce correct forms of otherwise correct words etc. [9] highlight the importance of manual flagging of errors (categorized into more linguistically motivated types) for system development.

We introduce a method of fully automatic analysis of translation errors. At minimum, our method requires the source, reference and hypothesis translations, i.e. nothing more than what is readily available in MT research. The implementation is language independent, but can take additional information into account, such as linguistic analyses (lemmatization, PoS tagging, synonym detection), training sets, dictionaries, etc.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 72–79, 2011.

| | |
|---|---|
| Source | The two remaining institutions also proved unable to reach an agreement. |
| Reference | Ani oba zbylé bankovní domy se tak nespojily. |
| cu-bojar | Zbývající dvě instituce také ukázalo, nelze dospět k dohodě. |

**Fig. 1.** Example of misleading reference: *bankovní domy* means ***banking** institutions*, a detail not present in the source and thus also not in the hypothesis (cu-bojar)

We evaluate the proposed method by comparing our automatically flagged errors with those identified manually in outputs of four English-to-Czech translation systems taking part in WMT09. The taxonomy of the manually flagged errors is the one of [17] – thus, in this work we design the method to find and classify errors in the taxonomy of this dataset, but the approach can be easily extended to other error types.

## 2 Method Description

Similarly to state-of-the-art approaches our method compares the hypothesis to a reference translation; this of course makes the approach sensitive to errors and liberal translations in the reference (see Figure 1 for an example of the reference falsely accusing a system of poor translation). Here we assume having a single reference translation, but the method can be easily extended to support several references – e.g. by greedily picking the reference that is most similar to the hypothesis.

Our goal is achieved in three steps: word alignment of the hypothesis and the reference, error detection and classification based on the alignment, and finally summarization of the discovered errors.

### 2.1 Word Alignment

The main difficulty in finding a word alignment between the hypothesis and reference is ambiguity, caused by frequently present repeated tokens (punctuation, particles), synonyms, words sharing the same lemma, but having different surface forms, etc. The aim is to resolve ambiguity to minimize the number of intersections between individual word alignments; we approach this problem by introducing a first-order Markov dependence for the alignments, stimulating adjacent words to be aligned similarly, which results in a preference towards aligning longer phrases.

The approach is very similar to bilingual HMM-based word alignment [18] in that hypothesis words are "emitted" by the hidden reference words. We assume a word-for-word correspondence (at most 1 link for any word) – in cross-language alignments, this assumption is not always viable, see e.g. [3] or [7], but here we need monolingual alignments. The search for the best alignment under these conditions has exponential time complexity, which is solved in this work via beam search.

However the main difference between our model and the one of [18] is that the emission and transition probabilities are hand-crafted – this way the model has the advantages of HMM-based word alignment, while not having to learn the models enables applying the model with the same result to sets of any sizes starting with single sentences.

The emission probability depends on the number of the same words in the hypothesis; for a word that occurs only once the probability equals 1 for matching reference words and 0 otherwise; for words that occur several times

$$p_{emit}(w_i^{(h)}|\emptyset) = \varepsilon,$$

$$p_{emit}(w_i^{(h)}|w_{a_i}^{(r)}) = \frac{(1-\varepsilon)\cdot[w_i^{(h)} = w_{a_i}^{(r)}]}{|\{w : w \in \text{hyp}, w = w_i^{(h)}\}|},$$

where $\varepsilon$ is a small constant. This allows repeating words to remain unaligned to make way for other, potentially better alignments of the same word in the hypothesis, while always aligning unique words to their counterpart.

The transition probabilities stimulate aligning the current word pair "in parallel" to the previously produced pair by penalizing the distance between the previous and the current reference word minus 1:

$$p_{trans}(w_{a_i}^{(r)}|w_{a_{i\_}}^{(r)}) \sim \exp(-b \cdot |a_i - a_{i\_} - 1|),$$

where $a_{i\_}$ is the index of the latest non-NULL alignment in the alignment $\mathbf{a}$.

In our work alignment is based on word lemmas – although this can increase the ambiguity in the alignment, it allows to detect wrong forms of a correctly picked lemma. In principle synonyms can be aligned in the same way using synonym detection; or, if no linguistic analysis is available, surface forms can also be used for alignment, but the number of unaligned words will naturally increase. Based on a very small sample of about 180 alignment points, our method reaches the recall of 74%, precision of 98% and alignment error rate of 84%.

### 2.2 Detecting Lexical Errors

Next the word alignment is used to classify the differences between the hypothesis and reference translations as different types of translation errors:

- unaligned words in the reference are marked as missing words; these are further classified into punctuation (`missP`), content (`missC`) and auxilliary (`missA`) words using POS tags
- unaligned words in the hypothesis are marked as untranslated if present in the source sentence (`unk`), and superfluous (`extra`) otherwise
- aligned words with different surface forms are marked as word form errors (`form`)

### 2.3 Detecting Order Errors

In this work the aligned words are in one-to-one correspondence[1], which enables calculating the common order similarity metrics (Hamming distance, Kendall's $\tau$ distance, Ulam's distance [2], Spearman's rank correlation coefficient, etc.) Here, however, we

[1] This can be ensured for other alignment methods by treating adjacent hypothesis words aligned to the same reference word as a single unit, as done by [16].

want to produce a more detailed analysis of the order errors, which would tell us which words are misplaced or switched. We approach this task by doing a breadth-first search for fixing the order in the aligned hypothesis words. The weighted directed tree for the search is such that

- there is one node per every permutation,
- there is an arc between two nodes only if the target node permutation differs from the source permutation by two adjacent symbols, whereas the relative order of the two symbols is wrong in the source and correct in the target node,
- the arc weight equals 1 in general; in order to enable block shifts, the arc weight is 0 when nodes contain adjacent transpositions – thus "continuing" to shift the same symbol in the same direction.

As a result switched word pairs are marked as short-range order errors (`ows`); a word shifted several positions towards the beginning or end of the sentence is marked as a long-range order error (`owl`).

### 2.4 Error Summarization

Marked translation errors are finally summarized on different levels, depending on the desired type of feedback on the machine translation system under evaluation. The highest level of detail is no summarization at all, enabling the developer to inspect the system output and the discovered errors sentence-by-sentence. This level of summarization is used in our work to calculate the precision and recall of every error type in comparison to manually tagged translation hypotheses.

Alternatively, in order to get a glimpse of the general properties of the translation the errors can be summarized by category, resulting in ratios of different types of erroneous words. Such output is similar to the tables of [17] and [4]; it can be used for qualitative comparison, enabling the developer to analyze the general weaknesses of translation systems.

Finally, at the lowest level of detail a linear combination of the ratios of error types can be used to score the system output as a whole.

## 3 Experiments and Results

In this section we compare the performance of our method to manually flagged errors in MT output; this is done both via the precision and recall of every error type, calculated against manual annotation.

### 3.1 Used Data

The reference dataset[2] consists of 200 sentences from the English-to-Czech WMT09 shared task. Tokens in outputs of four selected systems were manually tagged according to the Vilar taxonomy (e.g. `lex` or `form`). See [4] for more details on the dataset. The

[2] `http://ufal.mff.cuni.cz/euromatrixplus/downloads.html`

**Table 1.** Evaluation results: precision ($P$), recall ($R$) and F-score ($F$) of every error flag inside the corresponding group

| Wrong hyp. word | | | | Missing ref. word | | | | Misplaced word | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Flag | $P$ | $R$ | $F$ | Flag | $P$ | $R$ | $F$ | Flag | $P$ | $R$ | $F$ |
| extra | 0.102 | 0.842 | 0.181 | missC | 0.009 | 0.205 | 0.017 | ows | 0.130 | 0.337 | 0.187 |
| unk | 0.218 | 0.633 | 0.324 | missA | 0.038 | 0.445 | 0.070 | owl | 0.031 | 0.403 | 0.058 |
| form | 0.388 | 0.460 | 0.421 | **Punctuation** | | | | ops | 0.000 | 0.000 | 0.000 |
| disam | 0.000 | 0.000 | 0.000 | extraP | 0.281 | 0.793 | 0.414 | opl | 0.000 | 0.000 | 0.000 |
| lex | 0.000 | 0.000 | 0.000 | missP | 0.137 | 0.785 | 0.234 | | | | |

inter-annotator agreement is rather low (43.6% overall) probably due to differences in what the annotators think the correct output should be. Despite this shortcoming, we believe this is so far the only publicly available dataset of this kind.

Since each word of a hypothesis can have several flags (e.g. form and ows) we simplify the annotation by grouping the flags into four independent categories: wrong hypothesis words, missing reference words, misplaced words and punctuation. At most one flag from each category is allowed; conflicts in the manual annotations are resolved in favor of the automatically assigned flag. Every error flag is evaluated in the context of its group.

For most sentences, the dataset includes alternate markups from different annotators. Instead of resolving conflicts between the alternatives, we take the following strategy: for every sentence and every error group, the precision and recall are computed for every available markup independently; then, only the most similar (the one with the largest number of correct automatic flags) alternative is picked and used for the general evaluation. This type of evaluation is in line with manual annotation: each annotator is free to choose a slightly different "correct" version of the hypothesis and mark errors compared to this assumed wording. We allow our system to choose any of the possible annotations but require it to stick to it throughout the sentence.

### 3.2 Evaluation Results

Table 1 presents the individual precisions and recalls for every error type inside its category; for the sake of this evaluation the four translation hypotheses of our dataset were grouped together to produce a single score table. Some error types were not supported by our evaluation – phrase short- (ops) and long-range (opl) reordering, synonym disambiguation error (disam) and wrong lexical choice (lex) – which is why their precision and recall equal 0.

It can be seen that in comparison to human annotators, our evaluation marks many more words as errors – as a result the precision is mostly low while the recall is somewhat higher. In particular, since our method does not align synonyms and wrong translations of existing reference words, every disam and lex error is replaced with a pair of a missing reference word and a superfluous hypothesis word, which results in their high recall and low precision.

Recall of misplaced words is satisfactory; since alignment also influences their discovery, alignment with greater coverage would increase it significantly.

Overall, our precision and recall are still somewhat low but nevertheless comparable to the inter-annotator agreement on the dataset.

## 4 Related Work

For references to all the many automatic MT evaluation metrics please see e.g. [5]. Very few of these metrics go beyond a single score for the given test set.

[14] used morpho-syntactic information for automatically analyzing specific verb-related translation errors. [15] enriched the WER score by separately evaluating scores for individual parts of speech, allowing a finer comparison of MT systems but still providing too little information on actual errors made by the systems.

[13][3] implemented visualization of mismatches of up to two systems compared to the reference translation. Apart from that, probably the only implemented and published toolkit with the same goal is Meteor-xRay[4] [8]. Neither of these approaches tries to classify errors as we do.

[10] report an interesting idea where a large pool of the single-outcome metrics can be used to obtain a refined picture of error types the evaluated systems make. Decomposing such a global result down to the examples of errors is not as straightforward as with our approach.

A critical component of our system is the monolingual alignment between the reference and the hypothesis. Meteor-Xray uses the alignment algorithm underlying the Meteor metric but the aligning component could be shared with other MT applications, e.g. system combination [11], where fully unsupervised GIZA++ has been successfully used [12].

## 5 Future Work

The introduced approach can be developed in a number of directions. Most importantly the coverage of word alignment has to be increased to account for synonymous translations and incorrect translation attempts of existing reference words (`lex`). Alignments from GIZA++ and the METEOR metric and alignments mediated by the source sentence should be tested.

Secondly, evaluation with multiple references should be performed – although in practice such datasets are rare, they can cause much better agreement between manual and automatic annotations. Another implementation issue is supporting various methods of summarization, including producing a single score and inspecting errors sentence-by-sentence.

Word order errors could be related to automatic parses allowing to count misplaced phrases, not just words.

[3] That work was done as a part of the Failfinder project at the MT Marathon in Dublin; see `http://code.google.com/p/failfinder/` for the code.

[4] `http://www.cs.cmu.edu/~alavie/METEOR/`

## 6 Conclusions

We introduced a technique for automatic discovery and classification of types of errors in machine translation output. In principle it is language-independent but greatly benefits from (automatic) linguistic annotation.

We evaluated our method by comparing the outputs to the errors marked manually on a subset of English-to-Czech WMT09 sentences. While the precision and recall are still rather low, they are comparable to the inter-annotator agreement on the set.

We believe some kind of automated error analysis will soon become an inherent step in MT system development and that future developments of our proposed technique, especially the improvement in alignment, will increase the match with human annotation.

**Acknowledgements.** The work on this project was supported by the Czech Science Foundation grants P406/ 11/1499 and P406/10/P259 and the Estonian Science Foundation target financed theme SF0180078s08.

## References

1. Berka, J., Černý, M., Bojar, O.: Quiz-Based Evaluation of Machine Translation. Prague Bulletin of Mathematical Linguistics 95 (2011)
2. Birch, A., Osborne, M., Blunsom, P.: Metrics for mt evaluation: evaluating reordering. Machine Translation 24(1), 15–26 (2010)
3. Bojar, O., Prokopová, M.: Czech-English Word Alignment. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), pp. 1236–1239. ELRA (May 2006)
4. Bojar, O.: Analyzing Error Types in English-Czech Machine Translation. Prague Bulletin of Mathematical Linguistics 95 (2011)
5. Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., Zaidan, O.: Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR, pp. 17–53. Association for Computational Linguistics, Uppsala (July 2010), http://www.aclweb.org/anthology/W10-1703 (revised August 2010)
6. Callison-Burch, C., Koehn, P., Monz, C., Schroeder, J.: Findings of the 2009 Workshop on Statistical Machine Translation. In: Proceedings of the Fourth Workshop on Statistical Machine Translation, pp. 1–28. Association for Computational Linguistics, Athens (2009), http://www.aclweb.org/anthology/W/W09/W09-0401
7. DeNero, J., Klein, D.: Discriminative modeling of extraction sets for machine translation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1453–1463. Association for Computational Linguistics, Uppsala (July 2010), http://www.aclweb.org/anthology/P10-1147
8. Denkowski, M., Lavie, A.: Extending the METEOR Machine Translation Evaluation Metric to the Phrase Level. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 250–253 (2010)
9. Farrús, M., Costa-jussà, M.R., Mariño, J.B., Poch, M., Hernández, A., Henriquez, C., Fonollosa, J.A.R.: Overcoming statistical machine translation limitations: error analysis. In: Language Resources and Evaluation, pp. 1–28 (February 2011)

10. Giménez, J., Màrquez, L.: Towards heterogeneous automatic mt error analysis. In: ELRA, E.L.R.A. (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (May 2008)
11. He, X., Yang, M., Gao, J., Nguyen, P., Moore, R.: Indirect-hmm-based hypothesis alignment for combining outputs from machine translation systems. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 98–107. Association for Computational Linguistics, Stroudsburg (2008), http://portal.acm.org/citation.cfm?id=1613715.1613730
12. Matusov, E., Leusch, G., Banchs, R.E., Bertoldi, N., Dechelotte, D., Federico, M., Kolss, M., Lee, Y.S., Marino, J.B., Paulik, M., Roukos, S., Schwenk, H., Ney, H.: System Combination for Machine Translation of Spoken and Written Language. IEEE Transactions on Audio, Speech and Language Processing 16(7), 1222–1237 (2008)
13. Popel, M., Mareček, D.: unpublished (2010), http://code.google.com/p/failfinder/
14. Popovic, M., de Gispert, A., Gupta, D., Lambert, P., Ney, H., Mariño, J.B., Federico, M., Banchs, R.: Morpho-syntactic information for automatic error analysis of statistical machine translation output. In: Proceedings on the Workshop on Statistical Machine Translation, pp. 1–6, New York, USA (2006)
15. Popović, M., Ney, H.: Word error rates: decomposition over pos classes and applications for error analysis. In: Proceedings of the Second Workshop on Statistical Machine Translation, StatMT 2007, pp. 48–55. Association for Computational Linguistics, Stroudsburg (2007), http://portal.acm.org/citation.cfm?id=1626355.1626362
16. Tiedemann, J.: Word to word alignment strategies. In: Proceedings of COLING 2004, pp. 212–218, Geneva, Switzerland (2004)
17. Vilar, D., Xu, J., D'Haro, L.F., Ney, H.: Error analysis of machine translation output. In: Proceedings of the 5th LREC, pp. 697–702, Genoa, Italy (2006)
18. Vogel, S., Ney, H., Tillmann, C.: HMM-based word alignment in statistical translation. In: International Conference on Computational Linguistics, pp. 836–841, Kopenhagen, Denmark (1996)

# Automatic Word Sense Disambiguation and Construction Identification Based on Corpus Multilevel Annotation*

Olga Lyashevskaya[1], Olga Mitrofanova[2], Maria Grachkova[2], Sergey Romanov[2],
Anastasia Shimorina[2], and Alexandra Shurygina[2]

[1] NRU Higher School of Economics, Moscow
olesar@gmail.com
[2] St. Petersburg State University,
Universitetskaya emb. 11,
199034, St. Petersburg, Russia
alkonost-om@yandex.ru

**Abstract.** The research project reported in this paper aims at automatic extraction of linguistic information from contexts in the Russian National Corpus (RNC) and its subsequent use in building a comprehensive lexicographic resource – the Index of Russian lexical constructions. The proposed approach implies automatic context classification intended for word sense disambiguation (WSD) and construction identification (CxI). The automatic context processing procedure takes into account the following types of contextual information represented in the RNC multilevel annotation: lexical (lemma) tags (lex), morphological (grammatical) tags (gr), semantic (taxonomy) tags (sem), and combinations of the various types of tags. Multiple experiments on WSD and CxI are performed using RNC representative context samples for nouns. In each series of experiments we analyze (1) different context markers of meaning of target words and (2) constructions including context markers and target words.

**Keywords:** WSD, constructions, construction identification, Russian National Corpus, context classification.

## 1 Introduction

A research project currently underway at St. Petersburg State University in collaboration with the team of the Russian National Corpus (RNC [1,2,3]) contributes to two practical tasks: word sense disambiguation (WSD) in Russian texts and building a comprehensive lexicographic resource, the Index of Russian lexical constructions, based on corpus data. Formal representation of a construction here implies one or more target words surrounded by a set of slots. Slots constraints are

* This work was supported by Russian Foundation for Basic Research (grant No 10-06-00586) and the RAS Presidium program of basic research "Corpus Linguistics".

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 80–90, 2011.

formulated in terms of features available from corpus annotation layers, namely, lexical, morphological and semantic tags.[1] This is exactly how users use the corpus interface in order to look for this or that pattern associated with a particular word, e.g. *bog* 'God' + *s* 'with' + [noun or pronoun, instrumental case, human] for phrases like *nu i bog s nim* 'to heck with him/it'; *voz'mi* 'take.imper' + *i* 'and' + [verb, imperative] for phrases like *a on voz'mi da i skaži* 'and all of a sudden he said'; *glava* 'chapter' + [noun, genitive case, text] for collocations like *glava knigi/romana* 'a chapter of the book/novel', etc. Our approach assumes that many senses of polysemous words are entrenched in such constructions or, to put it differently, each sense can be matched to a list of partially prefabricated chunks where all traditional levels of language are taken into account. Thus Construction Grammar [5,6,7,8] and constructionally-oriented lexicographic resources meet the problem of word sense disambiguation (WSD).

WSD plays a crucial role in corpora development and use. A rich variety of reliable WSD techniques such as knowledge-(or rule-) based, statistical corpus-based WSD or their hybrids have been worked out and tested, see [9,10,11], among others. Knowledge-based WSD is performed with the help of semantic information stored in electronic lexicographic modules (e.g., WordNet [12], FrameNet [13]). Corpus-based WSD implies extraction and statistical processing of word collocations, which makes it possible to distinguish separate meanings of lexical items in context (e.g., [14,15], etc.). Hybrid WSD brings into action both lexical resources and corpus analysis (e.g., [16,17], etc.).

Richly annotated corpora prove to be valuable sources of linguistic evidence necessary for exploring word meanings, their interrelations, extracting lexical-semantic classes, developing taxonomies, etc. Statistical algorithms implemented in contemporary corpora processing tools ensure extraction of information on the frequency distributions of lexical, morphological and semantic markers. These data are indispensable for classification of word contexts and, thus, for proper identification of word senses in contexts [18,19].

Major WSD techniques were enabled in previous experiments on semantic ambiguity resolution in Russian texts. The use of lexical databases for Russian (e.g., an electronic thesaurus RuTes [20], the RNC semantic dictionary [21,22], RussNet lexical database [23]) provides good-to-high quality WSD. If lexicographic information is not available, statistical WSD techniques are indispensable in processing Russian texts. As experimental data have shown, it is possible to identify word meanings in contexts taking into account POS tag distributions [24] and lexical markers [25]; hybrid WSD occurs to be effective as well [26].

While WSD seems to be a well-elaborated methodology in NLP, methods and algorithms for identifying and extracting lexical constructions have been generally overlooked [27]. Until recently, constructions with partially filled lexical slots which either violate common syntactic laws or involve a certain degree of idiomaticity has been seen as a "pain in the neck" for parsing. Little effort has been made to extract patterns other than n-grams and fixed multi-word expressions. However, some recent research projects show an increased interest in the syntax-lexicon continuum [27,28];

[1] Information on syntactic dependencies is not used in our experiments since this kind of annotation is only available for the smaller part (1M) of the RNC [4].

cf., for example, a bottom-up approach to the compilation of hybrid constructions like “keep a [adj] eye on” or “keep a close [noun] on” (*keep a close/keen eye on, keep a close eye/watch on*) in StringNet [29].

Construction identification (CxI) in Russian is likely to be even more problematic from the computational point of view. Due to the so-called “free” word order and rich inflectional system, the use of n-grams here is not so effective as in English. Moreover, the elements of a construction can be easily omitted if they are mentioned in the previous context or inferable from the situation, so the existing dictionaries of valencies and other lexical resources on argument structure require considerable sophistication to use [26], not to mention the fact that a range of relevant information (especially for some parts of speech) may be missing.

Despite the complexity of the problem, we consider compiling the list of sense-distinctive lexical constructions a longer-term goal of the project. In this paper, we discuss how lexical, morphosyntactic and semantic information relevant for the selectional preferences of word senses can be reanalyzed as constructional patterns.

The paper is organized as follows. Section 2 describes data in use. Section 3 outlines the toolkit for collecting and processing data. In Sections 4 and 5 a number of experiments on statistical WSD for Russian nouns is analyzed. We address the results of the Targeted WSD (rather than All Words WSD) that has been run with regard to three types of contextual information: lexical markers (lemmas), morphological (grammatical) markers, semantic (taxonomy) markers – and compare the reliability of these WSD approaches. Some other preferential conditions such as context window size and training set size are also discussed. Section 6 presents possible application of the technique to the CxI method in order to provide data for the Index of Russian lexical constructions. Section 7 concludes.

## 2 Linguistic Data

Contexts for Russian nouns that refer to tangible objects and abstract notions serve as the empirical basis of the study. These include polysemous and/or homonymous words as *dom* ‘building, private space, family, etc.’; *organ* ‘institution, part of body, musical instrument, etc.’; *luk* ‘onion, bow’; *glava* ‘head, chief, cupola, chapter, etc.’; *vid* ‘view, form, document, image, verbal aspect, kind, species’; *ključ* ‘key, clue, clef, spring, etc.’; *sovet* ‘advice, council, etc.’; *ploščad’* ‘square, space, etc.’; *kosa* ‘braid, scythe, peninsula’, etc. Although the nouns used in experiments belong to different lexical-semantic groups, they reveal regular types of relations between meanings of polysemous words or between homonymic items. That is why the set of words in question is regarded as representative of noun class in general.

Sets of contexts were extracted from the Russian National Corpus, the largest annotated corpus of Russian texts containing about 400 M tokens. The texts included in the so called Main corpus of the RNC are supplied with three core types of annotation: (1) lemmas – lexical markers (canonical, dictionary forms of inflected words); (2) grammatical markers (morphosyntactic tagsets referring to POS and other inflectional grammatical features like case, gender, tense, etc.); (3) taxonomy markers (semantic tagsets referring to lexical-semantic classes). Taxonomy markers are available for the most frequent nouns, pronouns, adjectives, verbs and adverbs and represent a rather coarse-grained cross-classification of the lexicon (e.g. 'concrete', 'human', 'animal', 'space', 'construction', 'tool', 'container', 'substance', 'movement', 'part', 'diminutive', 'causative', 'verbal noun', and other lexical-semantic classes, cf. http://www.ruscorpora.ru/en/corpora-sem.html). Each word sense is formalized with a set of taxonomy markers, cf. *dom* 'house': 'concrete' + 'construction' + 'container'.

In the samples drawn from the corpus, lexical and grammatical tags are disambiguated and, to avoid a vicious circle, taxonomy markers assigned to a particular lexical item in a context represent its first, or central lexical meaning. For a target word, taxonomy markers stand for all of its recognized meanings.

As is well known, the frequency distribution of senses varies greatly from one word to another. In some cases it is highly skewed (there can be up to 90% uses of the first sense in a sample) whereas in other cases the distribution of senses is quite even. For the given nouns, we analyzed each word sense that is represented by 10 or more occurrences in the RNC. Word senses with fewer than 10 contexts in the corpus (such as *dom* 'common space' or *dom* 'dynasty') were excluded from the study. Manual disambiguation was performed for a training set of contexts for each word, the remaining part of ambiguous contexts was subjected to statistical WSD.

## 3 Toolkit for WSD and CxI

A Python-based toolkit was developed to perform Targeted WSD and CxI procedures. The toolkit performs (1) generation of context classes corresponding to particular meanings of a target word; and (2) generation of lists of the most frequent constructions where a particular meaning of a target word occurs.

The toolkit makes it possible to carry out linguistic and statistical analysis of contexts for polysemous words in various modes which are defined by the type of context markers to be taken into account in processing: lexical (lemma) tags (*lex*), morphological (grammatical) tags (*gr*), semantic (taxonomy) tags (*sem*), as well as combinations of various types of tags (*lex+gr*, *lex+sem*, *sem+gr*, *lex+sem+gr*). The toolkit also allows a researcher to specify certain parameters of experiments, such as (1) weight assignment to context items (if chosen, a weight value would be assigned to each item within the context); (2) context window size [$-l$; $+r$] (trivial values for $l$ and $r$ are allowed).

The WSD procedure is carried out in several stages. The first stage involves pre-processing of contexts in the experimental set $E$. Semantically and morphologically unambiguous or manually disambiguated contexts are selected randomly to form a training set $S$, while ambiguous contexts are treated as a trial set $T$. Machine learning is performed at the second stage. For each meaning of a word, its statistical pattern is established taking into account the frequencies of chosen context markers. Further, patterns of meanings, as well as trial contexts, are represented as vectors in a word space model. The third stage involves pattern recognition, i.e. selection of patterns nearest to vectors that correspond to ambiguous contexts. The Cosine measure was chosen as a similarity measure to perform word sense assignment. The Cosine measure was calculated by the following formula:

$$Cos(v_1,v_2) = |<v_1,v_2>|/(|v_1|\cdot|v_2|), \qquad (1)$$

where $<v_1,v_2>$ is the dot product of vectors $v_1$ and $v_2$. As a result, meanings triggered by particular patterns are automatically assigned to processed contexts on the basis of maximum similarity.

The resulting output includes the following information:

- attributed meaning with a similarity value for each context in the trial set *T*;
- precision (*P*): the percentage of contexts in the trial set *T* for which meanings were attributed correctly; it is possible to compute this rate if the operation was performed on the trial set *T* with manually resolved ambiguity;
- recall (*R*): percentage of valid contexts in the trial set *T*.

The CxI procedure requires co-occurrence data collected for context markers of word meanings. In our case, constructions are regarded as the most frequent combinations of a target word and certain tags which regularly occur within a certain left and/or right context and mark a given meaning of a target word. E.g., the target word *vid* that triggers the meaning 'kind' commonly occurs with frequent tags in the right context 'r:abstr der:v' (abstract noun derived from a verb: *vid dejatel'nosti* 'kind of activity', *vid straxovanija* 'kind of insurance', *vid obespečenija* 'kind of garantee'). Co-occurrence data are extracted from the training set *S*, so the accuracy of this information is in direct correlation with the size of the training set *S*. The resulting output includes the following information:

- a list of frequent constructions, i.e. combinations of a target word and relevant tags within left/right contexts;
- frequency data for each construction;
- lexical items which trigger meanings corresponding to semantic tags (*sem*) in constructions (also with frequencies of occurrence).

To run a WSD and CxI program, it is necessary to choose input files (a file with training contexts and a file containing trial contexts), select context window size, indicate whether weights should be assigned to context items, and choose the output file.

## 4 Conditions for WSD and CxI

Series of more than 1000 tests were carried out to establish the criteria for WSD and CxI: (1) to estimate the correlation between lexical, morphological and semantic tags, to compare the reliability of these criteria and to ascertain preferential conditions for their application; (2) to evaluate several parameters that can influence test results: context window size, proportional expansion of training sets of contexts for each meaning, etc. Evaluation of WSD quality was performed: the results of automatic WSD were compared with the results of manual WSD, precision *P* and recall *R* were calculated in all tests.

In the course of the experiments we studied the efficiency of context markers and their combinations: lexical (lemma) tags (*lex*), morphological (grammatical) tags (*gr*), semantic (taxonomy) tags (*sem*), combinations of various types of tags (*lex+gr*, *lex+sem*, *sem+gr*, *lex+sem+gr*). The highest efficiency was provided by a combination of lemma tags and semantic tags (*lex+sem*). The lowest efficiency was provided by isolated morphological tags (*gr*). However, a combination of all three types of tags (*lex+sem+gr*) and isolated lemma tags (*lex*) ensured good results as well [30,31].

Tests with variable context window size [*-l; +r*] ($l$, $r \leq 5$) were performed, so that the context window could be symmetric or asymmetric, and could be limited to a clause or a syntactic group. The context window size providing best results was defined in accordance with the values of precision ($P$) and recall ($R$). To combine precision and recall metrics, we used $F$-measure.

Parameters of the best context window were defined for different types of tags. Thus, if lexical tags of context elements are taken into account, the best context window size turns out to be [-4; +5]. When all three types of tags are combined, the best context window size proves to be [-3; +5]. The best context windows for the *lex+sem* tag combination are [-2;+4] and [-3; +4]: contexts of this size seem to be the most acceptable for nouns as they cover the most relevant information. In most cases this context window corresponds to the groups that stand in preposition to the target noun (adjectival groups) or in postposition (nominal, infinitival, etc. groups) and contain context items relevant for sense disambiguation. Context analysis with regard to syntactic relations showed an increase in precision ($P$) by 0.05…0.1.

Thorough analysis of contexts shows that the appropriate choice of the type of context markers and context window size alongside with expansion of the training set ($S$ = 100…500 contexts) ensures over 85% correct decisions on average ($P$≈0.85). Under such conditions, in series of experiments the number of correct decisions turned out to be no less than 50…60% ($P$≈0.50…0.60), in some cases up to 95…100% ($P$≈0.95…1). Experiments were performed with training sets of variable size $S$ = 10, 15, 55, 75, 100, 200, 500, … (up to all contexts except for those included in a trial set) and with proportional expansion of the training set $S$ being 10%, 15%, or 20% of $E$. According to our observations, the training set $S$ should contain at least 100 unambiguous contexts, while 500 contexts provide the best results. In general, to obtain reliable WSD results, the training set size $S$ should be no less than 20% of the experimental set size $E$. In other cases the proportion of correct decisions may be reduced because statistical patterns for meanings turn out to be rather ‘blurry’.

## 5 Identification of Context Markers for Russian Nouns

During the experiments we studied sets of contexts for Russian polysemous nouns extracted from the RNC. Context markers of several types were selected for each meaning of the target words. Much attention was paid to the lemma tags and semantic tags of the context items. For example, the target word *glava* ‘chief’ frequently co-occurs with the following lexemes forming its right context: *gosudarstvo* (‘state’ <r:concr t:space>), *federacija* (‘federation’ <r:concr t:space>), *region* (‘region’ <r:concr t:space pt:part pc:space>), *gorod* (‘city’ <r:concr t:space>), *fond* (‘fund’ <r:concr t:space pt:set sc:money>). These context markers can be combined to form a group of concrete nouns identifying space and place (<r:concr t:space>). To take another example, the target word *luk* ‘onion’ regularly co-occurs with such nouns as *ogurec* (‘cucumber’ <r:concr t:fruit t:food>), *orex* (‘nut’ <r:concr t:fruit t:food pt:part pc:plant>), and *kartoška* (‘potato’ <r:concr t:fruit t:food pt:aggr sc:fruit>). These nouns may be referred to as a group of concrete nouns denoting food. These examples show that the identification of context markers can be carried out not in terms of particular lexemes, but in terms of the lexical-semantic classes they belong to.

Context markers may differ not only in type, but also in the position they occupy with respect to a target word. Therefore, the right and left contexts of target words were examined separately. For instance, semantic tags indicating abstract nouns of perception (<r:abstr t:perc>) regularly occur in the right context of the target word *organ* (‘part of a body’). This fact allows us to consider them as context markers for the word in question. But when we explored the left context of the same word in the same meaning, we found out that other lexemes often serve as its context markers: e.g., adjectives, such as *čelovečeskij* ‘human’, *donorskij* ‘donor’ (<dt:hum>), nouns *zabolevanije*, *bolezn’* ‘disease’ (<t:disease>), etc. The context markers mentioned above are not to be found in any occurrences of the word *organ* in other meanings.

Experiments on WSD were also performed for polysemous words which reveal both independent and merged meanings. In case of the noun *dom*, the independent meanings are *m1a* ‘building’, *m1b* ‘private space’, *m2* ‘family’, *m3* ‘common space’, *m4* ‘institution’, *m5* ‘dynasty’. The merged meanings are formed by pairs (*m1a*/*m1b*, *m1a*/*m4*, *m1b*/*m2*) or triples (*m1a*/*m1b*/*m4*; *m1a*/*m1b*/*m2*) of independent meanings, which are almost indistinguishable in some contexts. Some markers extracted from the context allow us to predict the occurrence of merged meanings with high precision. For example, context markers of merged meaning *m1a*/*m4* are presented in such pairs as *detskij dom* ‘orphan’s home’, *invalidnyj dom* ‘home for disabled people’, *rodil’nyj dom* ‘maternity hospital’, *dom otdyxa* ‘holiday center’, *dom kino* ‘film theatre’, etc. The context marker that clearly indicates merged meaning *m1a*/*m1b*/*m2* can be found in such phrase as *hozjain doma* ‘host’. However, there are context markers that indicate purely independent meanings. For example, noun *roditel’* ‘tenant’ in *roditeli doma* ‘tenants are at home’ points out to meaning *m1a*. In some cases, such as *rodnoj dom* ‘home’, *roditel’skij dom* ‘one’s parent’s home’, the merged meaning *m1a*/*m1b* is more frequent than corresponding independent meanings.

# 6 Construction Identification

The data on a diverse breed of constructions and their productivity are much wanted by modern lexicography as well as in theoretical studies. A new generation of dictionaries poses a challenge to the computational learning of constructions: random examples illustrating this or that meaning of a word need to be replaced with the corpus-based and, to some extent, frequency-oriented lists of typical constructions and typical lexical fillers in these constructions [32]. The study of the “behavioral profile” [33] of a word in quantitative cognitive linguistics and works on acquisition of constructions also rely hugely on data annotated with a particular construction attributes.

Ch. Fillmore’s Constructicon [34] represents one possible way to design a constructionally-oriented database (on English corpus data). This project assumes that a limited group of constructions is picked up and annotated, with minor computer assistance, by the team of researchers. The question remains, does anybody know the approximate number of (lexical) constructions given that they are purely and inconsistently documented in existing linguistic resources and that any type of pattern starting from fixed expressions and up to very abstract structures like Adjective + Noun can be considered a construction? “Constructions are the rules that license ‘new’ linguistic signs based on other linguistic signs” [33:9], where signs are unique pairings of form and meaning, so the answer seems to be clear: this number approaches infinity.

Our approach is bottom-up seeding constructions where the main focus is on making generalizations upon individual constructions just as empirically-based theories of lexical acquisition suggest. These theories also hypothesize that generalizations can be done in various ways, and so our CxI approach assumes: any individual collocation can be associated with more than one generalized patterns.

Context markers play a crucial role in CxI. Lemma tags provide (partially) lemmatized collocations of high frequency. POS and other grammatical tags (such as case and infinitive) help to establish child – parent relations such as the following:

| | | |
|---|---|---|
| *po etomu.*[APRO].[dat] *povodu*<br>‘on this occasion’<br>*po takomu.*[APRO].[dat] *povodu*<br>‘on such an occasion’<br>*po dannomu.*[A].[dat] *povodu*<br>‘on this occasion’<br>*po malejšemu.*[A].[dat] *povodu*<br>‘at the slightest provocation’<br>… | ➔ | *po* .[A/APRO].[dat] *povodu*<br>‘on (some) occasion’ |
| *glava gosudarstva.*<t:space>.[gen]<br>‘a leader of a state’<br>*glava federacii.*<t:space>.[gen]<br>‘a leader of a federation’<br>*glava regiona.*<t:space>.[gen]<br>‘a leader of a region’<br>*glava goroda.*<t:space>.[gen]<br>‘a major of a city’<br>… | ➔ | *glava* .<t:space>.[gen]<br>‘a leader of (a region)’ |

**Fig. 1.** Child – parent relations in the constructions

The third type of generalization can be done with the help of taxonomy markers when lemmas are aggregated into the clusters with shared lexical semantics. In many cases, one or more lemmas are used most frequently in the clusters demonstrating effect of a strong prototype that attracts some weaker members.

As the results reported in Section 4 shows, generalized patterns give higher frequencies for the training data and thus provide better results in WSD. Figure 1 illustrates that generalizations are based on the repeated tags of individual constructions and that generalized patterns usually include tags of various levels. The adopted WSD approach provides us with the competing scores of individual tags and their combinations, and the CxI aim is to see in these messy numbers winning gestalt structures.

So far a number of CxI experiments have been done on a very simple class of constructions evoked by Russian nouns. A trivial cutting-off-tails technique has been exploited based on extracting and ranking *N* most frequent tags of each level in the WSD training samples. Since we are interested in sense-contrasting patterns, the following threshold principle is applied: not more than 10% examples of a given construction can be associated with other senses of a word.

The resulting lists of generalized patterns were tested against the similar lists of constructions compiled manually [25] and showed high levels of agreement (*P* is up to 80...90%). Recall levels (*R*) were significantly worse than control numbers; this can be explained by the small size of the training sets and the fact that human annotators used lexical groupings not available through the RNC taxonomy.

A future line of approach will presumably address more controversial verb argument structures and combine the use of the highest possible amount of data (namely, all contexts from the RNC pre-processed by the WSD script) with more sophisticated statistical techniques. With verb patterns, a new manually annotated resource [35] is planned to be used as a control set of data.

## 7 Conclusion

In this paper we discuss the Targeted WSD and bottom-up construction learning in Russian as two closely related tasks. Both approaches are based on context clustering and rely on three types of contextual information: lexical tags (lemmas); POS and other grammatical tags; semantic (taxonomy) tags.

The data under analysis involve the contexts of polysemous and/or homonymous Russian nouns extracted from the RNC. The following conditions for WSD were discovered: over 85% (in some cases up to 95%) correct decisions may be achieved through the use of Cosine measure, a training set varying from 100 up to 500 contexts that constitutes at least 20% of the experimental set *E*, context window size taking into account isolated lemma tags *lex* is [-4; +5], for combination of lemma tags and semantic tags *lex+sem* it is [-2;+4] and [-3; +4], for combination of three types of tags *lex+sem+gr* it is [-3; +5]. The highest efficiency was provided by a combination of lemma tags and semantic tags (*lex+sem*). Adding grammatical tags (*lex+sem+gr*) and isolating lemma tags (*lex*) ensured good results as well.

The following metaphor may be used to describe the difference between WSD and CxI methods. In image classification tasks, WSD can be compared with detecting the color spectrum of the images while CxI is compared with attempts to see the paths presumably embedded in the pictures. WSD deals with almost unstructured bags of features; CxI aims to gestalt a structure everywhere.

The proposed CxI approach involves bottom-up seeding the patterns that are typically associated with a target lemma. The main focus is on making generalizations upon individual constructions (collocations) taking into consideration the same tags (*lex*, *gr* and *sem*) as those used in WSD. The statistical approach to CxI is still under developement. The preliminary results addressing a very simple class of nominal constructions are encouraging but further exploratory work is required.

## References

1. Russian National Corpus, `http://ruscorpora.ru`
2. Russian National Corpus: 2003–2005. Indrik, Moscow (2005) (in Russian)
3. Russian National Corpus: 2006–2008. New results and future development. Nestor-Istorija, St. Petersburg (2009) (in Russian)
4. Nivre, J., Boguslavsky, I.M., Iomdin, L.: Parsing the SynTagRus Treebank of Russian. In: COLING 2008, Manchester, UK, vol. 1, pp. 641–648 (2008)
5. Goldberg, A.E.: Constructions. A Construction Grammar Approach to Argument Structure. University of Chicago Press, Chicago (1995)
6. Goldberg, A.E.: Constructions at Work: the Nature of Generalization in Language. Oxford University Press, Oxford (2006)
7. Fillmore, C.J.: The Mechanisms of Construction Grammar. Proceedings of the Berkeley Linguistic Society 14, 35–55 (1988)
8. Tomasello, M.: Constructing a Language: A Usage-Based Approach to Child Language Acquisition. Harvard University Press, Cambridge (2003)
9. Agirre, E., Edmonds, P. (eds.): Word Sense Disambiguation: Algorithms and Applications. Text, Speech and Language Technology, vol. 33. Springer, Heidelberg (2007)
10. Mihalcea, R., Pedersen, T.: Word Sense Disambiguation Tutorial (2005), `http://www.d.umn.edu/~tpederse/WSDTutorial.html`
11. Navigli, R.: Word Sense Disambiguation: a Survey. ACM Computing Surveys 41(2), 1–69 (2009)
12. WordNet, `http://wordnet.princeton.edu/`
13. FrameNet, `http://framenet.icsi.berkeley.edu/`
14. Pedersen, T.: A Baseline Methodology for Word Sense Disambiguation. In: Gelbukh, A.F. (ed.) CICLing 2002. LNCS, vol. 2276, p. 126. Springer, Heidelberg (2002)
15. Schütze, H.: Automatic Word Sense Disambiguation. Computational Linguistics 24(1), 23–97 (1998)
16. Leacock, C., Chodorow, M., Miller, G.: Using Corpus Statistics and WordNet Relations for Sense Identification. Computational Linguistics 24(1), 147–165 (1998)
17. Mihalcea, R.: Word Sense Disambiguation Using Pattern Learning and Automatic Feature Selection. Journal of Natural Language and Engineering 1(1), 1–15 (2002)
18. Mitrofanova, O., Panicheva, P., Lashevskaya, O.: Statistical Word Sense Disambiguation in Contexts for Russian Nouns Denoting Physical Objects. In: Sojka, P., et al. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 153–159. Springer, Heidelberg (2008a)
19. Mitrofanova, O., Lashevskaya, O., Panicheva, P.: Experiments on statistical WSD for Russian nouns in a corpus. In: Proceedings of the International Conference Corpora 2008, St. Petersburg, Russia, October 6–10, pp. 284–293 (2008b) (in Russian)
20. Lukashevich, N.V., Chujko, D.S.: Automatic WSD based on thesaurus knowledge. In: Internet-matematika 2007, Ekaterinburg, pp. 108–117 (2007) (in Russian)

21. Rahilina, E.V., Kobritsov, B.P., Kustova, G.I., Lashevskaja, O.N., Shemanaeva, O.J.: Semantic ambiguity as an application-oriented problem: word class tagging in the RNC. In: Computational Linguistics and Intellectual Technologies. Proceedings of the International Workshop Dialogue 2006, Moscow, pp. 445–450 (2006) (in Russian)
22. Kustova, G.I., Lashevskaja, O.N., Paducheva, E.V., Rakhilina, E.V.: Verb Taxonomy: From Theoretical Lexical Semantics to Practice of Corpus Tagging. In: Lewandowska, B., Dziwirek, K. (eds.) Cognitive Corpus Linguistics Studies. Peter Lang, Frankfurt (2009)
23. Azarova, I.V., Bichineva, S.V., Vakhitova, D.T.: Automatic WSD of the most frequent nouns (in terms of the structural units of RussNet). In: Proceedings of the International Conference Corpora 2008, St. Petersburg, Russia, October 6–10, pp. 5–8 (2008) (in Russian)
24. Azarova, I.V., Marina, A.S.: Computational context classification: preparing the data for the thesaurus RussNet. In: Computational Linguistics and Intellectual Technologies. Proceedings of the International Workshop Dialogue 2006, pp. 13–17. RGGU, Moscow (2006) (in Russian)
25. Kobritsov, B.P., Lashevskaja, O.N., Shemanajeva, O.J.: WSD in mass media texts: shallow rules and statistic evaluation. In: Internet–matematika 2005: Avtomaticheskaja obrabotka web-dannyx, Moscow, pp. 38–57 (2005) (in Russian)
26. Toldova, S.J., Kustova, G.I., Lashevskaja, O.N.: Semantic filters for WSD in the Russian National Corpus: verbs. In: Computational linguistics and intellectual technologies. Proceedings of the International Workshop Dialogue 2008, pp. 522–529. RGGU, Moscow (2008) (in Russian)
27. Sahlgren, M., Knutsson, O.: Workshop on Extracting and Using Constructions in NLP. In: NODALIDA 2009. SICS Technical Report T2009:10 (2009)
28. Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics, pp. 25–31, Los Angeles, CA (2010)
29. Wible, D., Tsao, N.-L.: StringNet as a Computational Resource for Discovering and Investigating Linguistic Constructions. In: Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics, Los Angeles, CA, pp. 25–31 (2010)
30. Lashevskaja, O., Mitrofanova, O.: Disambiguation of Taxonomy Markers in Context: Russian Nouns. In: Jokinen, K., Bick, E. (eds.) NODALIDA 2009. NEALT Proceedings Series, vol. 4, pp. 111–117 (2009)
31. Mitrofanova, O., Lyashevskaya, O.: Context markers of the nouns with concrete meaning in the lexico-semantic annotation of the RNC. In: XXXVIII International philological Conference, St. Petersburg (2009) (in Russian)
32. Atkins, B.T.S., Rundell, M.: The Oxford Guide to Practical Lexicography. Oxford University Press, New York (2008)
33. Gries, S.T., Divjak, D.: Behavioral Profiles: a Corpus-Based Approach to Cognitive Semantic Analysis. In: Evans, V., Pourcel, S.S. (eds.) New Directions in Cognitive Linguistics. John Benjamins, Amsterdam (2008)
34. Fillmore, C.J., Lee-Goldman, R.R., Rhodes, R.: The FrameNet Constructicon. In: Boas, H.C., Sag, I.A. (eds.) Sign-based Construction Grammar. CSLI Publications, Stanford (forthcoming)
35. Lyashevskaya, O.: Bank of Russian Constructions and Valencies. In: LREC 2010, pp. 1802–1805. ELRA (2010)

# Bootstrapping Bilingual Lexicons from Comparable Corpora for Closely Related Languages

Nikola Ljubešić[1] and Darja Fišer[2]

[1] Faculty of Humanities and Social Sciences, University of Zagreb
Ivana Lučića 3, 10000 Zagreb, Croatia
nikola.ljubesic@ffzg.hr
[2] Faculty of Arts, University of Ljubljana,
Aškerčeva 2, 1000 Ljubljana, Slovenia
darja.fiser@ff.uni-lj.si

**Abstract.** In this paper we present an approach to bootstrap a Croatian-Slovene bilingual lexicon from comparable news corpora from scratch, without relying on any external bilingual knowledge resource. Instead of using a dictionary to translate context vectors, we build a seed lexicon from identical words in both languages and extend it with context-based cognates and translation candidates of the most frequent words. By enlarging the seed dictionary for only 7% we were able to improve the baseline precision from 0.597 to 0.731 on the mean reciprocal rank for the ten top-ranking translation candidates with a 50.4% recall on the gold standard of 500 entries.

**Keywords:** bilingual lexicon extraction, cognates, comparable corpora.

## 1 Introduction

Bilingual lexicons are indispensable in most cross-lingual NLP applications and their compilation remains a major bottleneck in computational linguistics. Techniques for automatic extraction of translation equivalents from parallel texts have become well established [9] but since parallel corpora are scarce resources, especially for uncommon language pairs and domains, they often cannot be used. This is why an alternative approach has gained momentum in the past decade that relies on texts in two languages which are not parallel but comparable [2], [12] and therefore more readily available, especially from the increasingly rich web data [17].

The approach relies on the assumption that the term and its translation appear in similar contexts [2], [12], which means that a translation equivalent of a source word can be found by identifying a target word with the most similar context vector in a comparable corpus. However, a direct comparison of vectors in two different languages is not possible, which is why a dictionary is needed to translate the features of source context vectors and compute similarity measures on those. At this point we seem to be caught in a vicious cycle: the very reason why we are resorting to a highly complex comparable corpus approach for mining translation equivalents is the fact that we do not have a bilingual dictionary to use in the first place. This issue has largely remained unaddressed in previous research, which is why we propose a knowledge-light approach

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 91–98, 2011.

that does not require any bilingual resource. Instead, it takes advantage of similarities between the source and the target language in order to obtain a seed lexicon used for translating features of context vectors.

The paper is structured as follows: in the next section we give an overview of previous related work. In Section 3 we present the construction of the resources used in the experiment. Section 4 describes the experimental setup and reports the results of automatic and manual evaluation. We conclude the paper with final remarks and ideas for future work.

## 2 Related Work

Most research into bilingual lexicon extraction from non-parallel texts was inspired by [2] and [12] whose main assumption is that the term and its translation share similar contexts. The method consists of two steps: modeling of contexts and measuring similarity between the source-language and target-language contexts using a dictionary. The majority of approaches follow the bag-of-words paradigm and represent contexts as weighted collections of words using LL [3], TF-IDF [2] or PMI [16]. After word contexts have been built in both languages, the similarity between a source word's context vector and all the context vectors in the target language is computed using a similarity measure, such as cosine [2], Jaccard [10] or Dice [11].

Central to comparing context vectors across languages is the translation of features in context vectors, which assumes that a dictionary is available. Alternative solutions for situations when this is not the case have not been explored to a great extent but [6] show that it is possible to obtain a seed dictionary from identical and similarly spelled words. Slightly differently, [1] and [15] take advantage of transliteration rules for Arabic/Chinese to generate translation candidates, which is especially efficient for named entities and new vocabulary. At the subword level, [8] constructed string substitution rules to obtain cognates in Spanish and Portugese. As an addition to the standard approach, [13] use string similarity as a reranking criterion of translation candidates obtained with context similarity measures.

Our approach is closest to [6] in that we too use identical words as our seed dictionary with the difference that we iteratively extend the seed dictionary on every step and, since we are working with more similar languages, our extracted lexicon is of a higher quality and therefore more usable in a real-world setting.

## 3 Resources Used

### 3.1 Building a Comparable Corpus

In this experiment, we wish to extract translation equivalents for the general vocabulary. This is why we built a Croatian-Slovene comparable news corpus from the 1 billion-word hrWaC and the 380 million-word slWaC that were constructed from the web by crawling the .hr and .si domains [4]. We extracted all documents from the domains jutranji.hr and delo.si, which are on-line editions of national daily newspapers with a high circulation and a similar target audience. The documents were already tokenized, PoS-tagged and lemmatized, resulting in 13.4 million tokens for Croatian and 15.8 million tokens for Slovene.

### 3.2 Building a Seed Dictionary

Since no open-source machine-readable dictionary is available for Croatian and Slovene, we built a seed dictionary from the comparable news corpus by extracting all identical lemmas tagged with the same part of speech in both languages. With this, we exploit the strong similarity between Croatian and Slovene.[1] As Table 1 shows, the seed dictionary contains almost 33,500 entries, 77% of which are nouns. Manual evaluation of 100 random entries for each PoS shows that average precision of the dictionary is 72%, nouns performing the best (88%). The errors are mostly Croatian words found in the Slovene part of the corpus, probably originating from readers' comments which could be avoided by filtering the corpus by language on a sub-document level. There are also some false friends that probably cause more serious problems (e.g. *nesližben* which means *unofficial* in Croatian but *not part of sbd's job* in Slovene).

**Table 1.** Size and precision of the seed dictionary

| POS | Size | Precision |
|---|---:|---:|
| nouns | 25,703 | 88% |
| adjectives | 4,042 | 76% |
| verbs | 3,315 | 69% |
| adverbs | 435 | 54% |
| total | 33,495 | 72% |

### 3.3 Building a Gold Standard

For automatic evaluation and comparison of the results we built a small gold standard that contains 500 randomly selected nominal entries from a traditional broad-coverage Croatian-Slovene dictionary.

## 4 Experimental Setup

The goal of this experiment is to extract a bilingual lexicon from a comparable corpus with a seed dictionary of words from the corpus that are identical in both languages. We consider the translation equivalents obtained with this seed dictionary as baseline and then try to improve the results by extending the seed dictionary with contextually confirmed cognates and first translation candidates of the most frequent words. Throughout the experiment we are using best-performing settings for building and comparing context vectors as confirmed by our previous research [5]. Context vectors are built for all content words that appear in the corpus at least 50 times. The co-occurrence window is 7 content words, with encoded position of context words in that window, and Log-Likelihood as vector association measure. Vector features are then translated with the

[1] According to [14] the cosine for 3-grams in Croatian and Slovene of 74%. A similar similarity was computed for Czech-Slovak (70%) and Spanish-Portuguese (76%).

seed dictionary, after which Jensen-Shannon Divergence is used as a vector similarity measure. Finally, ten top-ranking translation candidates are kept for automatic and manual evaluation.

The evaluation measure is mean reciprocal rank. Although we extract translations for all content words, we report here the results of the automatic evaluation for nouns only due to space restrictions. In this experimental setup, recall is always 50.4% because we always find translations for 252 of the 500 nouns from the gold standard that satisfy the frequency criterion (50) in the source corpus and have at least one translation in the target corpus that meets the same frequency criterion. To calculate the baseline, we translated features in context vectors with the seed dictionary of identical words. Using the settings described above we achieve 0.597 precision for the baseline.

### 4.1 Adding Cognates to the Seed Dictionary

In this step of the experiment we augment the seed dictionary with cognates. They are calculated with BI-SIM [7], a modified bigram longest common subsequence function. The threshold for cognates has been empirically set to 0.7. First, translation equivalents are calculated as explained above taking into account 20 top-ranking translations. If we find a translation equivalent that meets the cognate threshold, we add that pair to the dictionary. We test dictionary expansion in two ways: by overwriting the existing dictionary entry with the identified cognate pair and by leaving the existing dictionary entry and disregarding the identified cognate pair.

**Table 2.** Manual evaluation of cognates

| POS | Size | Precision |
|---|---:|---:|
| nouns | 1,560 | 84% |
| adjectives | 779 | 92% |
| verbs | 706 | 74% |
| adverbs | 114 | 85% |
| total | 3,159 | 84% |

As Table 2 shows, we identified more than 3,000 cognates, almost half of which are nouns. Manual evaluation of 100 random cognates for each part of speech shows that cognate extraction performs best for adjectives (92%), probably because of the regular patterns used to form adjectives in Croatian and Slovene (e.g. Cro: *digital**an***, Slo: *digital**en***, Eng. *digital*).

It is interesting to see that the quality of the extracted cognates is 12% higher than the quality of the identical words. The reason for this is probably the contextual verification of cognates.

Table 3 contains the results of automatic evaluation of bilingual lexicon extraction with the seed dictionary that was augmented with cognates. Overwriting the existing dictionary entries with the new translation always performs better than leaving the old translation. By augmenting the seed dictionary with cognates, a 0.088 increase in precision is achieved.

**Table 3.** Automatic evaluation of bilingual lexicon extraction using the seed dictionary augmented with cognates (OW: existing entries were overwritten with cognate pairs, NOW: existing entries were kept)

| POS | Size | New | Precision-OW | Precision-NOW |
|---|---:|---:|---:|---:|
| baseline | 33,495 | 0 | 0.597 | 0.597 |
| cognates-N | 34,089 | 1,560 | 0.626 | 0.612 |
| cognates-Adj | 33,999 | 779 | 0.657 | 0.639 |
| cognages-V | 33,655 | 706 | 0.621 | 0.613 |
| cognates-Adv | 33,565 | 114 | 0.598 | 0.598 |
| cognates-NAdj | 34,593 | 2,339 | 0.679 | 0.641 |
| cognates-all | 34,823 | 3,159 | 0.685 | 0.649 |

### 4.2 Adding First Translation Candidates to the Seed Dictionary

In our previous research we showed that the precision of the first translation candidates of highly frequent words in the corpus was especially high [5]. We therefore decided to add to the seed dictionary the first translation candidates for words that appear in the corpus at least 200 times. If the seed dictionary already contains an entry, we again test dictionary expansion in the same two ways as described above.

Overall, first translation candidates yielded 1,635 more entries for the seed dictionary than cognates but their quality is much lower (by 21.5% on average). Almost 53% of the extracted first translation candidates are nouns, which are of the highest quality (71%) according to manual evaluation performed on a random sample of 100 first translation equivalents for each PoS. It is interesting to note that many of the manually evaluated first translation candidates were also cognates, especially among nouns (48%), which further strengthens the argument for using cognates in bilingual lexicon extraction tasks. The incorrect translation candidates were in 22.5% of the cases semantically closely related words, such as hypernyms, co-hyponyms or opposites that are not correct themselves but probably still contribute to good modeling of contexts and thereby helping bilingual lexicon extraction.

Table 5 gives the results of automatic evaluation of bilingual lexicon extraction with the seed dictionary that was augmented with first translation candidates. Again, overwriting the existing dictionary entries with the new translation outperforms leaving the old translation.

**Table 4.** Manual evaluation of first translation candidates for high-frequent words

| POS | Size | Precision | Cognates | Related |
|---|---:|---:|---:|---:|
| nouns | 2,510 | 71% | 48% | 9% |
| adjectives | 957 | 57% | 38% | 9% |
| verbs | 1,002 | 63% | 30% | 2% |
| adverbs | 325 | 59% | 26% | 4% |
| total | 4,794 | 62.5% | 35.5% | 6% |

**Table 5.** Automatic evaluation of bilingual lexicon extraction using the seed dictionary augmented with first translation candidates (OW: existing entries were overwritten with the extracted translation pairs, NOW: existing entries were kept)

| POS | Size | New | Precision-OW | Precision-NOW |
|---|---|---|---|---|
| baseline | 33,495 | 0 | 0.597 | 0.597 |
| 1st_trans-N | 33,964 | 2,510 | 0.662 | 0.625 |
| 1st_trans-Adj | 33,967 | 957 | 0.652 | 0.620 |
| 1st_trans-V | 33,695 | 1,002 | 0.641 | 0.609 |
| 1st_trans-Adv | 33,818 | 325 | 0.611 | 0.598 |
| 1st_trans-NAdj | 34,436 | 3,467 | 0.711 | 0.650 |
| 1st_trans-all | 34,817 | 4,794 | 0.714 | 0.651 |

When first translation candidates of all four PoS are added to the dictionary, precision is 0.117 over the baseline, outperforming cognates by 0.029. This suggests that first translation candidates of most frequent words have a greater impact on translating context vectors and on the quality of the extracted bilingual lexicon.

### 4.3 Combining Cognates and First Translation Candidates to Extend the Seed Dictionary

Finally, we combine the cognates and first translation candidates in order to measure the information gain obtained by applying both methods simultaneously. Since overwriting existing dictionary entries with new translation pairs consistently achieved better results than keeping the old ones, we only evaluate the former setting here. An additional goal of this experiment is to check which information is more beneficial for extracting translation equivalents from a comparable corpus without an external dictionary, cognates or first translation candidates. This is why in one version of the seed dictionary cognates were added first and then first translation candidates (enabling cognates to be overwritten by translation equivalents) while the second version was built the other way around (enabling translation equivalents to be overwritten by cognates).

When we prefer first translation candidates over cognates, we achieve precision of 73.1% while changing the preference gives a slightly lower score of 72.3%. This shows that first translations are more beneficial for the context vector translation procedure even when this information is combined.

Manual evaluation of a random sample of 100 translation equivalents we extracted with the best-performing augmented seed dictionary shows that 88 contained the correct translation among the ten top-ranking translation candidates. In the first position 64 of those were found and 24 in the remaining nine positions. What is more, many lists of ten top-ranking translation candidates contained not one but several correct translation variants. Also, as many as 59 of correct translation candidates were cognates, suggesting that the results could be improved even more by a final re-ranking of translation candidates based on cognate clues.

## 5 Conclusions and Future Work

In this paper we presented a knowledge-light approach to bilingual lexicon extraction from comparable corpora of similar languages. It outperforms related approaches both in terms of precision (0.731) and recall (50.4%). Unlike most related approaches it deals with all content words, and enriches the seed dictionary used for translating context vectors from the results of the translation procedure itself. The proposed approach is directly applicable on a number of other similar language pairs for which there is a lack bilingual lexicons due to socio-economic reasons.

In the future, we wish to refine the methods for building the comparable corpus. We are also looking into possibilities to extend the approach in such a way that it will be able to handle multi-word expressions as well because they are an important component for most HLT tasks. And, last but not least, we wish to address polysemy by refining the translation procedure of context vectors as well as measuring similarity of contexts within and across languages.

**Acknowledgments.** Research reported in this paper has been supported by the ACCURAT project within the EU 7th Framework Programme (FP7/2007-2013), grant agreement no. 248347, and by the Slovenian Research Agency, grant no. Z6-3668.

## References

1. Al-Onaizan, Y., Knight, K.: Translating Named Entities Using Monolingual and Bilingual Resources. In: ACL 2002, pp. 400–408 (2002)
2. Fung, P.: A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. In: Farwell, D., Gerber, L., Hovy, E. (eds.) AMTA 1998. LNCS (LNAI), vol. 1529, pp. 1–17. Springer, Heidelberg (1998)
3. Ismail, A., Manandhar, S.: Bilingual lexicon extraction from comparable corpora using in-domain terms. In: COLING 2010, pp. 481–489 (2010)
4. Ljubešić, N., Erjavec, T.: hrWaC and slWaC: Compiling Web Corpora for Croatian and Slovene. In: Proceedings of the 3rd International Workshop on Balto-Slavonic Natural Language Processing, Plze, Czech Republic, (September 1–5, 2011)
5. Fišer, D., Ljubešić, N., Vintar, Š., Pollak, S.: Building and using comparable corpora for domain-specific bilingual lexicon extraction. In: Proceedings of the 4th ACL-HLT Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web, Portland, Oregon, USA, (June 24, 2011)
6. Koehn, P., Knight, K.: Learning a translation lexicon from monolingual corpora. In: ULA 2002, pp. 9–16 (2002)
7. Kondrak, G., Dorr, B.J.: Identification of Confusable Drug Names: A New Approach and Evaluation Methodology. In: COLING 2004 (2004)
8. Markó, K., Schulz, S., Hahn, U.: Multilingual Lexical Acquisition by Bootstrapping Cognate Seed Lexicons In: RANLP 2005, pp. 301–307 (2005)
9. Och, F.J., Ney, H.: Improved Statistical Alignment Models. In: ACL 2000, pp. 440–447 (2000)
10. Otero, P.G., Campos, J.R.P.: An Approach to Acquire Word Translations from Non-parallel Texts. In: Bento, C., Cardoso, A., Dias, G. (eds.) EPIA 2005. LNCS (LNAI), vol. 3808, pp. 600–610. Springer, Heidelberg (2005)

11. Otero, P.G.: Learning Bilingual Lexicons from Comparable English and Spanish Corpora. In: MTS 2007, pp. 191–198 (2007)
12. Rapp, R.: Automatic identification of word translations from unrelated English and German corpora. In: ACL 1999, pp. 519–526 (1999)
13. Saralegi, X., San Vicente, I., Gurrutxaga, A.: Automatic Extraction of Bilingual Terms from Comparable Corpora in a Popular Science Domain. In: BUCC 2008 (2008)
14. Scannell, K.P.: Language similarity table, http://borel.slu.edu/crubadan/table.html
15. Shao, L., Ng, H.T.: Mining New Word Translations from Comparable Corpora. In: COLING 2004 (2004)
16. Shezaf, D., Rappoport, A.: Bilingual Lexicon Generation Using Non-Aligned Signatures. In: ACL 2010 pp. 98–107 (2010)
17. Xiao, Z., McEnery, A.: Collocation, semantic prosody and near synonymy: a cross-linguistic perspective. Applied Linguistics 27(1), 103–129 (2006)

# Combining Topic Specific Language Models

Yangyang Shi, Pascal Wiggers, and Catholijn M. Jonker

Man-Machine Interaction Group, Delft University of Technology
Mekelweg 4, 2628CD, Netherlands
shiyang1983@gmail.com

**Abstract.** In this paper we investigate whether a combination of topic specific language models can outperform a general purpose language model, using a trigram model as our baseline model. We show that in the ideal case — in which it is known beforehand which model to use — specific models perform considerably better than the baseline model. We test two methods that combine specific models and show that these combinations outperform the general purpose model, in particular if the data is diverse in terms of topics and vocabulary. Inspired by these findings, we propose to combine a decision tree and a set of dynamic Bayesian networks into a new model. The new model uses context information to dynamically select an appropriate specific model.

## 1 Introduction

A language model should capture the regularities in a language so that it can judge the fluency of a sequence of words. Statistical language models implement this notion of fluency as a probability that is assigned to every word sequence.

The $n$-gram language model, that conditions the probability of a word on the previous $n$ words, is still the most popular language model. It has the advantages that the necessary statistics are easy to collect and that it captures much of the local regularity of language. As it only takes a local context into account, the model is also robust to distortions, a feature that is most useful if a sentence hypothesis processed by the language model may contain incorrect words, as is for example the case in speech recognition.

However, these models fail to capture important long range dependencies between words. Most attempts to improve the $n$-gram language model focus on modelling a larger part of the context, by taking a larger part of the history into account or by explicitly including syntactic [1] or semantic relations [2,3,4,5].

An assumption behind $n$-gram models, but also behind many other language models, is that the relative frequency of a word combination is the same for all situations. This is a useful assumption to ensure reliable parameter estimates, but it is clearly incorrect. For example, in this particular paper, the word sequence “language model” is much more likely than in an average text.

An alternative approach would be to create dedicated models for particular contexts, for example, a model trained on papers on language modeling might be a more accurate model for the text in this paper than a general purpose language model trained on a large, diverse set of texts. Such an approach leads to two questions: 1) do dedicated models indeed outperform general purpose models in the proper context? 2) if so, how can one select a dedicated model for a given situation.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 99–106, 2011.

In this paper, we investigate these two questions. To answer the first question we compared the performance of a trigram language model trained on our whole data set with the performance of dedicated models trained on specific parts of the data set. Our analysis is presented in Section 3. In Section 4 we turn to the second question. We present two language models based on dynamic Bayesian networks (DBNs) that combine multiple dedicated models. In Section 5 we propose a new adaptive model, called DBN tree, that builds upon these DBN models. This model dynamically selects a language model based on contextual information. We start with a brief overview of related work in section 2.

## 2 Related Work

The idea of combining multiple specific models has been used before in language modelling [6]. For example [7] interpolate multiple specific $n$-grams to obtain $n$-gram estimates. The mixture models of [8] combine multiple topic specific models at the sentence level, using weighted interpolation. Each of their component models is interpolated with a general $n$-gram model to handle data sparsity problem. The dynamic Bayesian network based language model of [4] is an extension of this work. They argue that sentence level mixture models can outperform other language models because they capture coherence in a text.

In terms of modeling techniques our models are closely related to those Bayesian network models and to the decision tree based language models of [9]. Although decision trees by themselves do not outperform the simpler $n$-grams in language modeling, they are a valuable component of more sophisticated models, such as random forrest language models [10].

## 3 Comparing Specific and General Models

The hypothesis behind our work is that models specifically created for a particular domain outperform more general models on the same domain. For this to be true, the advantage of modeling specific syntactic and semantic patterns in the data has to outweigh the data sparsity that comes with a smaller data set.

We tested this hypothesis on the Corpus Spoken Dutch (Corpus Gesproken Nederlands; CGN)[11,12]. An 8 million word corpus of contemporary Dutch spoken in Flanders and Netherlands. This data set is made up of 15 components, each with its own 'genre'. The different components range from spontaneous conversations over dinner, over political discussion to broadcast news (Table 1).

For all experiments discussed in this paper, we randomly selected 80% of the data in every component for training, 10% for development testing and 10% for evaluation. We created a vocabulary with 44368 words, which contains all unique words that occur more than once in the training data. All words in the data that are not in the vocabulary were replaced by an out-of-vocabulary(OOV) token. Using all training data, we created an interpolated trigram model:

$$\hat{p}(w_i|w_{i-2}w_{i-1}) = \lambda_1 p(w_i|w_{i-2}w_{i-1}) + \lambda_2 p(w_i|w_{i-1}) + \lambda_3 p(w_i), \tag{1}$$

where $w_i$ is the $i$-th word in a sentence. Each of the components of this model was estimated using maximum likelihood. The interpolation weights $\lambda_1, \lambda_2$ and $\lambda_3$ were estimated using Expectation-Maximization on the development test set. In the same way, we created a specific model for 14 components of the CGN corpus[1] We tested every component model on the corresponding component specific evaluation set and the general model on each of the evaluation sets in turn. Table 1 shows perplexity results per component of the evaluation set. Perplexity is calculated according to:

$$PP(w_1 w_2 \ldots w_t) = 2^{-\frac{1}{t}\log P(w_1 w_2 \ldots w_t)}, \qquad (2)$$

where $w_1 w_2 \ldots w_t$ is the data in the evaluation set.

**Table 1.** Comparison in terms of perplexity of specific models and general models per component of the CGN data set

| component | specific models | general models | | |
|---|---|---|---|---|
| | | trigram | topic | cluster |
| a (spontaneous face-to-face) | 220.85 | 222.83 | 226.31 | 221.37 |
| b (interviews) | 196.95 | 214.39 | 213.20 | 212.00 |
| c (spontaneous telephone) | 186.56 | 189.19 | 193.57 | 188.22 |
| d (spontaneous telephone) | 189.90 | 194.45 | 192.14 | 193.28 |
| e (business negotiations) | 110.27 | 153.35 | 154.11 | 152.15 |
| f (interviews broadcast) | 274.59 | 284.32 | 283.47 | 281.73 |
| g (debates) | 287.19 | 372.34 | 366.95 | 349.62 |
| h (lessons) | 298.08 | 315.01 | 314.07 | 313.44 |
| i (live commentaries) | 275.65 | 425.24 | 402.42 | 369.33 |
| j (news reports) | 345.37 | 369.28 | 367.52 | 361.41 |
| k (news) | 366.14 | 573.01 | 560.47 | 563.87 |
| l (interviews broadcast) | 425.70 | 440.11 | 435.14 | 434.42 |
| n (lectures) | 398.80 | 444.08 | 436.57 | 434.55 |
| o (read book texts) | 573.59 | 705.90 | 682.52 | 695.76 |
| overall | - | 277.67 | 274.08 | 272.50 |

The second column of Table 1 shows that the perplexity of the specific component models is lower than the perplexity of the general trigram model for every component. For some components the difference is relatively small. This is especially the case for the components that contain spontaneous speech, e.g. component a. For these components we may expect little performance improvement from a model that is based on component specific submodels. For others, most notably the live commentaries, the news, and the read texts the difference is considerable. Based on these results, we may hypothesize that the general model looses performance on those components that contain a diverse mix of topics and and a more diverse vocabulary. Put differently, in those cases it fails to model the exceptions to the rule.

[1] We left out component m, as it it too small to create reliable models with.

# 4 Bayesian Network Models

The results in the previous section suggest that we can improve upon the general trigram model if we can exploit the information in the component models. Previous work [8,4,5] suggests that we can do so by treating the component as an explicit variable in the language model. As the value of this variable is unknown during use, it will be inferred from the previous words. Dynamic Bayesian networks provide a convenient formalism to express such models in. Before presenting our models we will briefly discuss dynamic Bayesian networks.

## 4.1 Dynamic Bayesian Networks

Bayesian networks are a method for reasoning with uncertainty that combine probability theory and graph theory [13]. 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 all variables given by:

$$\prod_V P(V|Parents(V)), \tag{3}$$

where $V$ is a random variable. Dynamic Bayesian networks (DBNs) model processes that evolve over time [14]. They consist of a Bayesian network that defines the relations between variables at a particular time step and a set of arcs that specify how the variables depend on previous time steps. Several efficient inference algorithms have been developed for DBNs [15]. Training is usually done with an instance of the Expectation-Maximization (EM) algorithm.

## 4.2 A Topic-Based Model

We implemented a socalled topic based Bayesian network model, in which the CGN components are the topics. Figure 1c shows the first three time slices of the corresponding network. As in the trigram, every word $w_i$ depends on the previous two words. As before we use an interpolated model that combines trigram, bigram and unigram statistics. The interpolation weights were set using a held out development test set. In addition, each of these components depends on the topic variable $T$, that takes as its values the 15 components of the CGN. $E$ signals the end of a sentence and $EOS$ the end of an utterance. The latter is included to ensure that the model is a proper language model in the sense that the sum of the probabilities of all possible word sequences is 1. $N$ is a helper variable that counts the words in a sentence. It is used to make sure that a word is not conditioned on words in a previous sentence.

We first trained this model on the complete data set with a uniform prior for the component variable. As a result every component model corresponds to the general trigram model. We then trained the model on the complete training set, but this time the component variable was set to the correct component value for every document. The models resulting from this run were interpolated with the models of the previous run to ensure that every model covered the same complete vocabulary. This way, the OOV rate is same for all models.

As before we evaluated the model on every component evaluation set in turn. In this case the component variable was treated as a hidden variable, i.e. the model does not know the component it is dealing with beforehand, but has to infer this value. Table 1 shows the results. With the exception of components a, c and e the model performs slightly better than the general trigram model.

### 4.3 A Cluster-Based Model

The topic model is based directly on the components of the CGN. However, these components may not form coherent subsets of the data, which makes it difficult for the model to infer which type of data it is dealing with. As discussed above, it is likely that the components that contain spontaneous speech are rather similar, while the data in other components such as news and live commentaries is more diverse. To take this into account, we reclustered the training data. For this we represented every document as a weight vector. The length of the vector corresponds to the number of different semantically salient lemma types in the vocabulary that were found by removing all function words and common content words from the vocabulary. We used lemmas rather than words, as inflections are not important for topicality. The entries of the vectors are weights that indicate the relation between the document and the lemmas. We used term frequency-inverse document frequency (TF-IDF) weights as widely used in information retrieval [16]:

$$\mathrm{weight}(i,j) = \begin{cases} (1+\log(\mathrm{tf}_{ij}))\log(\frac{N}{\mathrm{df}_i}) & \mathrm{tf}_{ij} > 0, \\ 0 & \mathrm{tf}_{ij} = 0, \end{cases} \tag{4}$$

where $N$ is the number of documents and the term frequency $\mathrm{tf}_{ij}$ counts the number of times lemma $i$ occurs in document $j$. High frequency lemmas are thought to be characteristic for the document. Higher counts reflect more saliency of a word for a document but the scale is not linear. Observing a word twice as much does not mean that it is twice as important. Therefore, term frequencies are logarithmically scaled. This quantity is weighted by the inverse document frequency $\mathrm{df}_i$ which gives the number of different documents lemma $i$ occurs in. The idea is that lemmas that occur in many documents are semantically less discriminating. This component is also logarithmically weighted.

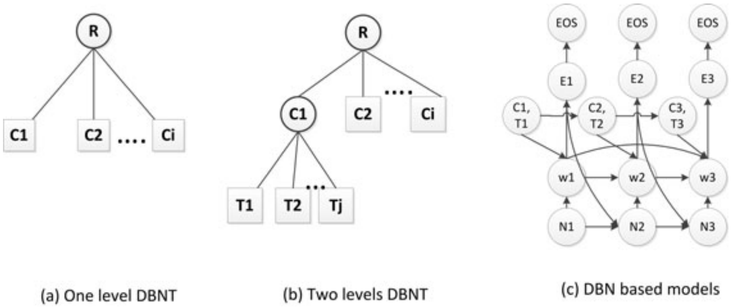
Together the word vectors span a high-dimensional space in which each dimension corresponds to a lemma. To measure semantic similarity between documents we applied a cosine metric. Clustering was done using $k$-means clustering, with $k = 16$. Every document was annotated with its cluster number, after which we trained a topic based model as descibed above.

Table 1 shows that the model performs better than the general trigram model on all components. With the exceptions of components d, k and o it also outperforms the topic-based model based on all components. The improvement is highest for the live commentaries and debates, components on which the trigram model did considerably worse than the baseline set by the component specific models. For the other components the improvements are smaller, but as mentioned above, it should be noted that for several components there is not that much to gain compared to the component specific models.

It is likely that better results can be obtained by optimizing the number of clusters in the model to ensure that each cluster does form a coherent, sufficiently large, data set. However, the computational demands of the DBN based models quickly increase as a function of the number of clusters as it sums over all cluster models to arrive at a probability for a word in every time step. Furthermore, this sum of component probabilities implies that the perplexity of the combined model will typically be higher that that of an appropriate specific model. Therefore, we propose a new type of model that selects a single specific model in every time step in the next section.

## 5 The Dynamic Bayesian Network Tree Model

A Dynamic Bayesian Network Tree (DBNT) combines a set of dynamic Bayesian network language models and a decision tree. The result is a language model that can change its parameters and structure depending on the context. For every word in a sentence, the decision tree is used to select one of the component language models based on available context information, such as the word history. Each of the nodes of the tree has an associated DBN language model. Figure 1a shows an example of a one-level DBNT. The components of this model can for example be the component specific models of section 3. More specific models are put in lower branches of the tree, to ensure that these models generalize sufficiently, they are interpolated with the more general models along the path from the root to their leaf. For example, every component model in Figure 1a is interpolated with the general model $R$ in the root. While model $T1$ in the two level DBNT shown in Figure 1b is interpolated with the root model $R$ and the intermediate model $C1$.



(a) One level DBNT

(b) Two levels DBNT

(c) DBN based models

**Fig. 1.** Two simple dynamic Bayesian network trees and a DBN based model. (a) One level DBNT. (b) Two level DBNT, with one nested clustering procedure. (c) The DBN based models, $W_i$ represents the word at the $i$-th position, $E$ signals whether current word is the end of a sentence, $EOS$ signals whether current word is the end of a sequence, $N$ gives word position in a sentence, and $T, C$ are the topic and cluster variable, respectively.

### 5.1 Learning DBNT Language Models

The learning procedure of the DBNT models is a two stage process. In the first step the decision tree is learned and used to split the training data into smaller subsets, where subsets on the same level of the tree do not overlap and node lower in the tree subdivide the data of their parent into smaller sets. For this any decision tree induction algorithm can be used. Alternatively, a hierarchical clustering procedure can be used. In the second stage, a component DBN model is trained for every subset in the data using standard DBN algorithms. The structure of these models is predetermined. To learn the interpolation weights for the models along a path in the model the EM algorithm [17] or re-estimation method [8] can be used.

### 5.2 Selection of Component Language Models in Prediction

The idea behind DBNT language models is to dynamically apply different component language models in different contexts. Any function that selects an appropriate model based on contextual information can be used. In particular, one can use the probability assigned by every component model to (part of) the word history to select the model that will predict the next word in the sentence.

## 6 Conclusion

We showed that specific language models outperform general language models if the specific model fits the data. We might improve our general models, if we can exploit the knowledge of the specific language models in a general model. Based on ideas in [8,4,5,16], we created and tested two DBN based models: a general topic model and a general cluster model, that make use of the knowledge of specific language models. The results showed that both models perform better than general trigram models. However, the computational demands of the DBN based models quickly increase as a function of the number of clusters or topics. Therefore, we proposed a new type of model that selects a single specific model in every time step. Our future work is to implement and test this new model.

## References

1. Chelba, C., Jelinek, F.: Exploiting syntactic structure for language modeling. In: Proceedings of the 17th International Conference on Computational Linguistics, vol. 1, pp. 225–231. ACL, Stroudsburg (1998)
2. Rosenfeld, R.: A maximum entropy approach to adaptive statistical language modelling. Computer Speech and Language 10, 187–228 (1996)
3. Schwenk, H.: Efficient training of large neural networks for language modeling. In: Proceedings IEEE International Joint Conference on Neural Networks, 2004, vol. 4, pp. 3059–3064 (2004)
4. Wiggers, P., Rothkrantz, L.: Combining topic information and structure information in a dynamic language model. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 218–225. Springer, Heidelberg (2009)

5. Shi, Y., Wiggers, P., Jonker, C.: Language modelling with dynamic bayesian networks using conversation types and part of speech information. In: The 22nd Benelux Conference on Artificial Intelligence, BNAIC (2010)
6. Clarkson, P., Robinson, A.J.: Language model adaptation using mixtures and an exponentially decaying cache. In: Proc. ICASSP 1997, Munich, Germany, pp. 799–802 (1997)
7. Kneser, R., Steinbiss, V.: On the dynamic adaptation of stochastic language models. In: Proceedings of ICASSP 1993, Minnapolis(USA), vol. II, pp. 586–589 (1993)
8. Iyer, R., Ostendorf, M., Rohlicek, J.R.: Language modeling with sentence-level mixtures. In: HLT 1994: Proceedings of the Workshop on Human Language Technology, pp. 82–87. Association for Computational Linguistics, Morristown (1994)
9. Bahl, L.R., Brown, P.F., de Souza, P.V., Mercer, R.L.: A tree-based statistical language model for natural language speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 37, 1001–1008 (1989)
10. Xu, P., Jelinek, F.: Random forests in language modeling. In: Proceedings of EMNLP, pp. 325–332 (2004)
11. Hoekstra, H., Moortgat, M., Schuurman, I., van der Wouden, T.: Syntactic annotation for the spoken dutch corpus project (cgn). In: Computational Linguistics in the Netherlands 2000, pp. 73–87 (2001)
12. Oostdijk, N., Goedertier, W., Eynde, F.V., Boves, L., Pierre Martens, J., Moortgat, M., Baayen, H.: Experiences from the spoken dutch corpus project. In: Proceedings of the Third International Conference on Language Resources and Evaluation, pp. 340–347 (2002)
13. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
14. Dean, T., Kanazawa, K.: A model for reasoning about persistence and causation. Computational Intelligence 5, 142–150 (1989)
15. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, University of California, Berkeley (2002)
16. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, Reading (1999)
17. 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, 1–38 (1977)

# Czech HMM-Based Speech Synthesis: Experiments with Model Adaptation*

Zdeněk Hanzlíček

University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics,
Univerzitní 8, 306 14 Plzeň, Czech Republic
zhanzlic@kky.zcu.cz

**Abstract.** This paper describes some experiments on model adaptation for statistical parametric speech synthesis for the Czech language. For building an experimental TTS system, HTS toolkit was utilised. Speech was represented by using high-quality analysis/synthesis system STRAIGHT. For definition of speech unit context, a new reduced set of contextual factors was proposed. During model clustering, some missing contextual factors, that were not included in this set, can be simulated by using combined context-related clustering questions. The model transformation was performed by a combination of CMLLR and MAP adaptation. Speech data from 3 male and 3 female speakers was used in our experiments. In the performed listening test, speech generated from regularly trained and adapted models was compared. Both voices were evaluated as identical and of a similar quality.

**Keywords:** HMM-based speech synthesis, speaker adaptation.

## 1 Introduction

Nowadays, statistical parametric (HMM-based) speech synthesis [1] is one of most researched synthesis methods. A great advantage of this method is the possibility to generate new voices by an adaptation of models trained for another speaker or even of models trained by using data from several different speakers [2]. Many adaptation methods have been developed – for an overview see [3].

Some basic experiments on HMM-based speech synthesis applied to the Czech language was already presented in [4]. This paper describes some consecutive experiments on model adaptation, which was performed by using a combination of 2 methods: CMLLR (constrained maximum likelihood linear regression) and additional MAP (maximum a posteriori) adaptation.

* This work was supported by the Grant Agency of the Czech Republic, project No. GAČR 102/09/0989 and by the Technology Agency of the Czech Republic, project No. TA01011264. Author would also like to thank Prof. Hideki Kawahara from Wakayama University for his permission to use the STRAIGHT analysis/synthesis method [5]. The access to the MetaCentrum computing facilities, provided under the programme “Projects of Large Infrastructure for Research, Development, and Innovations” LM2010005 funded by the Ministry of Education, Youth, and Sports of the Czech Republic, is appreciated.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 107–114, 2011.

For speech representation the analysis/synthesis method STRAIGHT [5] was utilized. Our experimental TTS system was built by using the well-known HTS toolkit [7]. Prosodic and linguistic characteristics of particular language are captured in a rich context of context-dependent units (models). Speech or language properties taken into account are called contextual factors. For the Czech language, we define a new reduced set of those factors.

For a more robust model parameter estimation, models are clustered using a decision tree-based context clustering algorithm. This process is controlled by simple context-related questions. To compensate the absence of some factors in our reduced set, we proposed the combined clustering questions which test the combinations of more factors.

Speech produced by regularly trained and adapted models was compared in a listening test. Speech data from 3 male and 3 female speakers were used. Results showed that speech produced by adapted models nearly of the same quality as speech generated from regularly trained models. Moreover, both voices were evaluated as identical.

The paper is organized as follows. In Section 2 a description of our HMM-based speech synthesis system and its settings are presented. Experimental evaluation is presented in Section 3. Finally, Section 4 summarizes the paper and outlines our future work.

## 2 System Overview

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

- **STRAIGHT** (Speech Transformation and Representation based on Adaptive Interpolation of weiGHTed, version 4.0) [6]
- **SPTK** (Speech Signal Processing Toolkit, version 3.3) [8]
- **HTS** (HMM-based Speech Synthesis System, version 3.4.1) [7]

### 2.1 Training Stage

Training stage can be roughly divided into 3 main parts:

1. **Parameter extraction** – Speech signal was sampled at 16 kHz. STRAIGHT analysis method used Gaussian $F_0$ adaptive window with 5 ms shift. Composed parameter vector contained 40 mel cepstral coefficients, $\log F_0$ value and 5 band aperiodicity coefficients, again with their delta and delta-delta.
2. **Model training** – Model parameters were estimated from speech data by using maximum likelihood criterion. We employed 5-state left-to-right MSD-HSMM with single Gaussian output distributions. First, robust models for particular monophones are trained. Then, context-dependent models are derived and retrained. The definition of unit context for our experiments is described in Section 2.4.
3. **Context clustering** – For a more robust model parameter estimation, context clustering based on MDL (Minimum Description Length) criterion was performed. Decision trees were separately constructed for cepstral, $\log F_0$, aperiodicity and duration parts of models.

### 2.2 Adaptation

A lot of modification and combination of basic adaptation methods (MLLR and MAP) have been developed – see e.g. [3]. For our experiments, we select CMLLR (constrained maximum likelihood linear regression) combined with additional MAP (maximum a posteriori) adaptation.

During the adaptation, multiple linear transformation functions are estimated by using speech data from target speaker. Since it is not possible to estimate a transform for each (clustered) context-dependent model, each transform is usually shared by a group (class) of related models.

In our experiments, classes sharing one transform were derived from context-independent units, i.e. models containing the same central monophone were adapted by using the same transform.

### 2.3 Synthesis Stage

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 models which are not available (were not trained). The final speech waveform is reconstructed from the generated parameters by using STRAIGHT-based vocoding.

### 2.4 Contextual Factors

In the HMM-based speech synthesis method, the phonetic and prosodic characteristics of a given language are respected by the specification of so called contextual factors. A speech unit (and the corresponding model) is given as a phone with its phonetic and prosodic context information. In this manner, the language prosody is modelled implicitly – in various contexts different units/models can be used. The context description is usually very rich [9]. Contextual factors are mostly defined as

- the position of the current phone/syllable/word in the parent syllable/word/ phrase
- the length of the current syllable/word/phrase in phones/syllables/words etc.

For a richer context description, the prosodic properties of speech are more precisely captured in the models. On the other hand, for a greater amount of contextual factors and wider range of their values, more training data is necessary to ensure a sufficient occurrence of all feasible combinations of defined factors.

In [4] a basic set of contextual factors was proposed. Contrary to other languages [9], this set was quite reduced. Based on some informal listening test, a new set of contextual factors was designed.

It encompasses the following prosodic features:

- *Prosodic clause* – a linear segment of speech delimited by pauses.
- *Prosodeme* – a rather abstract unit describing communication function. In the Czech language, it is usually connected with the last prosodic word in the phrase. In our experiments, only 4 main prosodeme types were distinguished: terminating satisfactorily (TS), terminating unsatisfactorily (TU), non-terminating (NT) and a formal null prosodeme.

- *Prosodic word* – a group of words belonging to one stress, often considered as a basic rhythmic unit.

For a more detailed description of Czech prosody see e.g. [10]. Compared to the default factor set proposed in [4], information on syllable boundaries was excluded from our new set. Our informal experiments revealed a low influence of that information. Moreover, the syllabification is ambiguous for the Czech language, in some cases syllable margins cannot be strictly determined. Thus, most contextual factors based on syllables would not be very precise. A more thorough study on the significance of particular contextual factors is planned to be performed in the future.

The set of contextual factors is summarized in Table 1. All factors, that define positions within the prosodic structure, were limited to values between 1 and 4, further positions are denoted 5+. The main motivation for this is the presumption that only several first marginal positions are prominent. Positions deeper inside the units become less distinguishable and the accurate position determination is supposed to be irrelevant.

A context-depended unit (or model) can be represented by a string

$$a_1–a_2+a_3\text{@P:}b_1\_b_2\text{@W:}c_1\_c_2/d_1$$

where all subscripted lower case letters are contextual factors defined in Table 1. The other characters in this string help to refer to particular factors (e.g. during model clustering).

**Table 1.** Contextual factors

| Factors | | Possible values |
|---|---|---|
| $a_1, a_2, a_3$ | Previous, current and next phoneme | Czech phoneme set (see e.g. [4]) |
| $b_1, b_2$ | Phone position in prosodic word (forward and backward) | 1, 2, 3, 4, 5+ |
| $c_1, c_2$ | Prosodic word position in clause (forward and backward) | |
| $d_1$ | Prosodeme type | TS, TU, NT, null |

### 2.5 Combined Context-Related Clustering Questions

The clustering algorithm utilises predefined set of context-related questions to build a decision-tree. By definition of more complex questions, some contextual factors, that were not included in the basic set, can be partly simulated, e.g.

- Phone position in the clause can be compensated by a combination of *phone position in the prosodic word* and *prosodic word position in the clause*. Obviously, only several marginal positions (related with the first and last word in the clause) can be reasonably defined this way. However consistently with position values definition in Table 1, only the marginal positions are prominent and are beneficial to be determined accurately.

- *Prosodic word length in phones* can be determined by combination of forward and backward *phone position in the prosodic word*. The accurate length can be determined only for words of length 1–4 phones. For longer words, at least one of position takes the value 5+, therefore the accurate length cannot be determined from contextual factors of one unit. However, the marginal values are expected to be most important again. *Clause length in prosodic words* can be expressed analogously.

A more thorough study on using such composed context-related questions for model clustering will be performed in the future. In our current experiments, the aforementioned simple combinations were employed.

# 3 Experiments and Results

## 3.1 Experimental Data Description

For our experiments speech data from 6 different speakers were utilised. This data was originally recorded for the purposes of a unit selection TTS system [11]. Thus, the overall quality was guaranteed. For simplification, male speakers are denoted $M_{AJ}$, $M_{JS}$, $M_{TF}$ and female speakers $F_{KI}$, $F_{MR}$, $F_{PP}$ – subscripted letter are initials of their names.

For our experiments, we selected one hour of speech from each speaker. Though the quality rises with the amount of training data, our previous experiments [4] showed that one hour of data is enough for synthesised speech of an acceptable quality.

Since recorded utterances from speakers $M_{AJ}$, $M_{JS}$, $F_{KI}$ and $F_{MR}$ were equal, we decided to use those speakers for training of initial models for adaptation. Thus, the differences in results for particular speakers should be caused by the variance between their voices and not by the selection of training utterances. Data from remaining speakers were employed for adaptation.

For pragmatic reasons, were preferred adaptation between speakers of the same gender, i.e. female-to-female or male-to-male. No cross-gender adaptation was performed in our experiments.

Equal utterances from all speakers should be also ideal for training of an average voice [2] – both gender dependent and independent. However, our informal experiments did not reveal any noticeable improvement – probably more data from more speakers should be used. Thus, we decided for a simpler experimental setup with one-speaker initial models. More thorough average voice experiments are planned in the future.

## 3.2 Quality Evaluation

In the first test, the quality of speech synthesised from adapted HMMs was evaluated. Two main questions were inspected

1. Naturally, the overall quality of speech generated from adapted models is expected to be lower when compared to speech synthesised from regularly trained models. How significant is the quality degradation caused by the adaptation process?

2. In case the amount and quality of speech data is sufficient, a regular model training could be performed. How distinctive is the difference between speech trained and adapted from the same amount of data?

To answer the first question, 1 hour of pure speech data (i.e. computed without pauses) was used to train models for particular speakers. Then, models were adapted by 10 minutes of speech from another speaker. In the test, participants listened to pairs of utterances generated from models

- trained from 1 hour of data from speaker X
- trained from 1 hour of data from another speaker and adapted with 10 minutes of data from speaker X

Listeners should select an utterance which of higher quality. The following 5-point scale was used:

1. utterance A is better than B
2. utterance A is slightly better than B
3. both utterances are similar
4. utterance A is slightly worse than B
5. utterance A is worse than B

The results are presented in upper half of Table 2. Utterances synthesised from regularly trained models were mostly evaluated as equal or slightly better. However, some listeners nearly continually preferred the adapted voice.

10 minutes of speech, that were utilised for the adaptation, could be also used for an independent training of a new model set. Naturally, no quality results could be expected from such a low amount of training data. However, we wanted to know whether the quality will be really poor or still acceptable. Thus, we synthesised pairs of utterances by using models

- trained from 10 minutes of data from speaker X
- trained from 1 hour of data from another speaker and adapted with 10 minutes of data from speaker X

Those pairs of utterances were inserted into the aforementioned test. The results are presented in the lower part of Table 2. Speech synthesised by using 10 minutes of training data was mostly evaluated as slightly worse than speech generated from adapted models. However, some utterances were evaluated as really worse and some as qualitatively equal.

### 3.3 Voice Identity Evaluation

Within the HMM-based speech synthesis framework, the only purpose of model adaptation is a change of voice identity. Thus, the similarity between the adaptation data and the synthesised speech . However, the comparison between regularly trained and adapted models is maybe more useful, because these are two main alternatives for obtaining a new voice identity. Since we wanted to prove that voice obtained by model

**Table 2.** Comparison of speech quality

| Compared utterances (A – B) | | Score |
|---|---|---|
| regular training | adaptation (10 minutes) | (mean ± std) |
| $M_{TF}$ (1h) | $M_{AJ} \rightarrow M_{TF}$ | $2.91 \pm 0.83$ |
| | $M_{JS} \rightarrow M_{TF}$ | $2.46 \pm 0.67$ |
| $F_{PP}$ (1h) | $F_{MR} \rightarrow F_{PP}$ | $2.75 \pm 0.72$ |
| | $F_{KI} \rightarrow F_{PP}$ | $2.88 \pm 0.66$ |
| $M_{TF}$ (10m) | $M_{AJ} \rightarrow M_{TF}$ | $3.88 \pm 0.86$ |
| | $M_{JS} \rightarrow M_{TF}$ | $3.97 \pm 0.73$ |
| $F_{PP}$ (10m) | $F_{MR} \rightarrow F_{PP}$ | $4.01 \pm 0.71$ |
| | $F_{KI} \rightarrow F_{PP}$ | $3.76 \pm 0.89$ |

adaptation is close to the voice obtained by regular training, we decided for a preference test with corresponding setup.

11 participants took part in our test. They listened to 12 pairs of utterances and evaluated their similarity according the following scale

1. both voices are identical
2. voices are very similar
3. voices are slightly similar
4. voices are totally different

Again 1 hour of speech was used for training of initial models and 10 minutes for their adaptation. The results are presented in Table 3. Notation $M_X$ or $F_X$ means, that results are calculated for both male ($M_{AJ}$ and $M_{JS}$) or female ($F_{MR}$ and $F_{KI}$) speakers. In most cases, both voices were evaluated as equal or rarely as very similar.

**Table 3.** Comparison of voice identity

| Compared utterances | | Score |
|---|---|---|
| regular training (1 hour) | adaptation (10 minutes) | (mean ± std) |
| $M_{TF}$ | $M_X \rightarrow M_{TF}$ | $1.26 \pm 0.31$ |
| $F_{PP}$ | $F_X \rightarrow F_{PP}$ | $1.18 \pm 0.25$ |

## 4 Conclusion and Future Work

In this paper, first experiments on speaker adaptation for HMM-based speech synthesis for the Czech language were presented. For building an experimental TTS system, HTS toolkit was utilised. For speech representation, STRAIGHT analysis-synthesis methods was used.

Our experiments proved, that for a small amount of training data (10 minutes) the adaptation of a different speaker's model set is preferable to the regular training of a fully new model set. Moreover, listening tests also revealed that speech generated from adapted models is of a similar quality as speech produced by models regularly trained by the same amount of speech data as the initial models for the adaptation. The difference in voice identity was also appreciated as insignificant.

In our future experiments, we will mainly focus on using speech data which is generally problematic to employ in the speech synthesis, e.g. data recorded in a common environment by non-professional speakers. Our aim is to be able to utilise speech data uttered in a spontaneous style. This would significantly increase the amount of voices that could be synthesised.

## References

1. Zen, H., Tokuda, K., Black, A.W.: Review: Statistical parametric speech synthesis. Speech Communication 51, 1039–1064 (2009)
2. Yamagishi, J., Usabaev, B., King, S., Watts, O., Dines, J., Tian, J., Hu, R., Guan, Y., Oura, K., Tokuda, K., Karhila, R., Kurimo, M.: Thousands of Voices for HMM-Based Speech Synthesis. In: Proceedings of Interspeech 2009, pp. 420–423 (2009)
3. Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., Isogai, J.: Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm. IEEE Transactions on Audio, Speech, and Language Processing 17, 66–83 (2009)
4. Hanzlíček, Z.: Czech HMM-Based Speech Synthesis. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS (LNAI), vol. 6231, pp. 291–298. Springer, Heidelberg (2010)
5. 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. In: Speech Communication, vol. 27, pp. 187–207 (1999)
6. STRAIGHT, a speech analysis, modification and synthesis system, http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTadv/index_e.html
7. HMM-based Speech Synthesis System (HTS), http://hts.sp.nitech.ac.jp
8. Speech Signal Processing Toolkit (SPTK), http://sp-tk.sourceforge.net
9. Tokuda, K., Zen, H., Black, A.W.: An HMM-based Speech Synthesis System Applied to English. In: Proceedings of IEEE Workshop on Speech Synthesis, pp. 227–230 (2002)
10. 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)
11. 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)

# Effective Parsing Using Competing CFG Rules

Miloš Jakubíček

Natural Language Processing Centre, Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
jak@fi.muni.cz

**Abstract.** In this paper a new pruning method for a rule-based parser is described that relies on separating the underlying grammar rules into several mutually competing levels. This method has been developed and exploited for Czech in the syntactic parser `Synt` to reduce the number of possible output derivation trees. The algorithm behind operates on a so called packed forest of trees, a compressing data structure used for internal representation of parallel analyses, and thus performs very effectively. An evaluation of its contribution has been performed on the Brno Phrasal Treebank showing that the algorithm significantly prunes the resulting tree space while preserving perspective parses.

## 1 Introduction

One of the main problems of parsing still remains: how to choose the best resulting parse tree from a possibly large set of candidates. This applies to rule-based parsing as well as statistical parsers, and all combinations of those. Moreover, applying deep-knowledge methods to each tree in the form of an exhaustive search is often not possible due to the size of the search space. The pain of this task is closely bound to what the parser relies on: a grammar or a treebank.

To address this problem in case of a statistical parser (see e. g. [1] for Czech), one basically faces two issues: how to create a reliable treebank and how to create such one that will be large. Similarly, in case of rule-based parsers, we stand in front of a grammar that we want to be small enough so that we can maintain it and that should achieve large coverage while not overgenerating too much. Clearly such mutually contradictory constraints are hardly achievable within a pure context-free grammar (CFG). Neither the probabilistic extension of this formalism (PCFG) is solely satisfying – while it is definitely a reasonable approach, since manual assignment of grammar rule probabilities has only very limited validity, it brings in the same issues a statistical-only parser has (the need for data that are hard to obtain in quality and size) and as such does not fully solve the problem.

On the other hand, we can further take the advantage of a rule-based parser to say that there are rules capturing various syntactic phenomena that do not only differ in how probable they occurrence is, but are actually mutually exclusive: one of them might be applied only after another one has failed. This can be implemented by separating all grammar rules into different precedence levels and filtering parse trees by the best achieved level, effectively forcing the CFG rules on different levels to compete with each other.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 115–122, 2011.

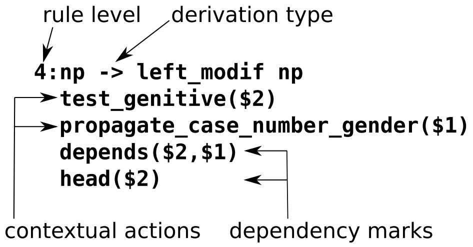
In further text the implementation of an effective algorithm is presented that exploits this idea in case of the Czech parser Synt whose main components are briefly described in the next section. Furthermore an evaluation of this method has been performed showing that it can significantly reduce the number of resulting derivation trees but doesn't harm parsers precision.

## 2 Syntactic Parser synt

Synt [2,3,4] is a rule-based syntactic parser for Czech with a CFG backbone. To address the very serious problem of grammar maintenance and development, the grammar is edited in the form of a metagrammar that contains only 240 rules. From this metagrammar, a full grammar is automatically generated by applying simple transformations that are associated with each rule and mostly regard the fact that Czech is a free-word-order language (a sample transformation creates a permutation of all right-hand side non-terminals in a rule or generates clause rules from a clause pattern with corresponding derivation type). A full grammar contains almost 4,000 rules.

### 2.1 Grammar

Each grammar rule may be also given a rule level indicated by an integer – the higher number is assigned to the rule, the lower is the rule priority. Currently, the whole grammar is structured into 7 levels, 0 being the default and most prioritized level. Finally, a grammar rule might be associated with dependency marks that are used to build a dependency graph and thus enabling also dependency output of the parser (although the primary and most developed output is the phrasal one). A sample rule is shown in Figure 1.



```
4:np -> left_modif np
  test_genitive($2)
  propagate_case_number_gender($1)
  depends($2,$1)
  head($2)
```

**Fig. 1.** A sample grammar rule illustrating key rule features

### 2.2 Chart and Forest of Values

The analysis done by Synt [5, p. 77] proceeds in two main steps: first, a basic CFG analysis is performed, using a head-corner chart parser, a variant of a general chart parsing as described by [6]. In this part, the internal chart structure (a multigraph) is built and serves as the core result for next steps.

To capture contextual phenomena, the resulting chart is subject to multiple contextual actions that mostly handle morphological agreement and other lexical specifics.

This allows to separate a clean and simple CFG analysis from more complex techniques that, while mostly expressable within a CFG formalism, would significantly complicate the grammar[1]. During the evaluation of these actions, a new data structure is built and represents main result of the whole analysis – a so called *forest of values*, a packed representation of all syntactic trees where each node (*a value*) contains related morphological information. The relation between both of the structures is demonstrated in Figures 2 and 3.

From this forest all other outputs are retrieved, including $n$ best syntactic trees or unambiguous syntactic structures [7]. A forest of values can be viewed as a graph of value nodes where each node holds multiple lists of children, with each list representing one possible analysis. The packing is achieved by sharing same children (list items) among all the lists. To get all possible derivation trees, one would recursively traverse the graph and pick a single list with children in each node.

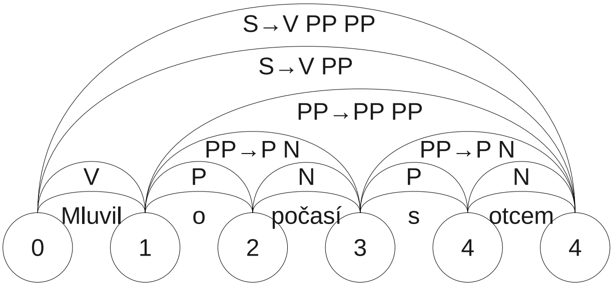


**Fig. 2.** A sample chart structure representing the result of CFG parsing of the sentence: *Mluvil/(he) talked/V o/about/P počasí/weather/N s/with/P otcem/father/N*

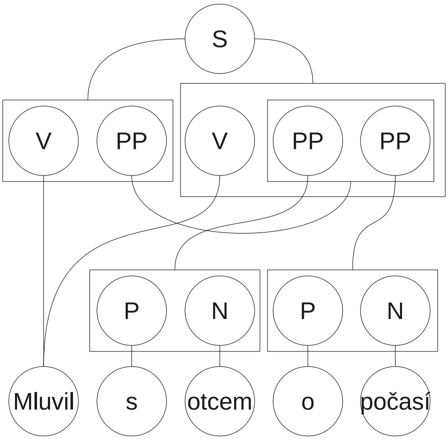


**Fig. 3.** A sample forest of values built from the chart given in Figure 2

[1] While these actions are in-programmed and as such Turing-complete.

# 3 Pruning by Rule Levels

## 3.1 Chart-Based Local Pruning

First implementation of the metagrammar design and parsing algorithms in Synt used a naive algorithm for pruning of the chart based on rule levels at the local scope of a single chart edge: if a chart edge had multiple parent edges with different rule levels, only the parents with lowest level have been preserved, all others have been removed. This was however, only sufficient when the difference in rule levels was present within the locality of the given edge, as is illustrated in Figure 4. Non-local differences were not propagated upwards in the chart and hence situations like the one demonstrated in Figure 5 remained uncovered by this approach.

Another important disadvantage of this method was that it actually harmed the coverage of the parser to some extent due to the fact that the pruning was performed on the chart before computing contextual actions and building the forest of values – in rare (but possible) cases it happened that a pruned analysis would have passed the contextual actions while all others that were preserved in the chart after the pruning have failed in the contextual actions.

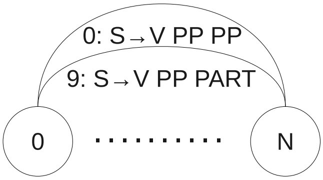


**Fig. 4.** A positive example of local pruning on the chart (rules are prefixed by their levels). Here the S → V PP PART edge would get removed.

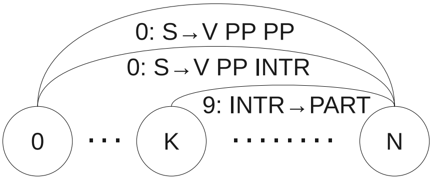


**Fig. 5.** A negative example of local pruning on the chart (rules are prefixed by their levels). No pruning would be performed because the rule level does not distinguish in the local scope.

## 3.2 Forest-Based Non-local Pruning

The new algorithm that has been implemented and is presented in this paper addresses both of the issues mentioned above – it performs non-local pruning so that it effectively filters all but best-leveled analyses and it operates on the forest of values instead of the chart so that it cannot cause the parsing process to fail even though some analysis was possible.

**Algorithm 1.** Forest-based non-local pruning algorithm pseudocode

```
Require: $V$ – value node, $CL$ – list of children to be considered
  $newChildLevel = 0$;
  for all $C \in CL$ do
    $L = \text{MAX}(C_e, C_c)$
    if $L \geq newChildLevel$ then
      $newChildLevel = L$
    end if
  end for
  if $V_c \geq newChildLevel$ then
    removeAllChildrenLists($V$)
    addChildrenList($V$, $CL$)
    $V_c = newChildLevel$
  else if $V_c \leq newChildLevel$ then
    return
  else
    addChildrenList($V$, $CL$)
    $V_c = newChildLevel$
  end if
```

Moreover, the pruning process happens on-the-fly when the forest of values is being built up from the chart. Henceforth the algorithm doesn't slow the analysis, but on contrary speeds it up because the resulting forest of values is significantly smaller.

The whole pruning algorithm is described in pseudo-code below and illustrated in Figure 6 and proceeds as follows: each value node $N$ in the forest of values contains the rule level of the corresponding edge in the chart $N_e$ and maximum rule level of all its children $N_c$. Before adding a new possible analysis (i. e. another list of children) it retrieves the maximum $C_{max}$ of the rule levels of all the new children, where a level of child $M$ is defined as $max(M_e, M_c)$. If this new level $C_{max}$ is greater than the current children level $N_c$, this analysis is discarded. If $C_{max}$ is smaller than $N_c$, all of the current analyses are substituted by this new one and $C_{max}$ is updated to $N_c$. Otherwise (when $C_{max} = N_c$) this analysis is just added to the current list of children.

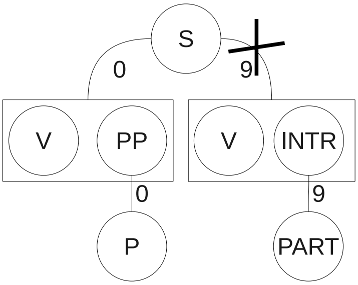


**Fig. 6.** Forest-based non-local pruning: rule levels get propagated (see Figure 7) and prune the structure

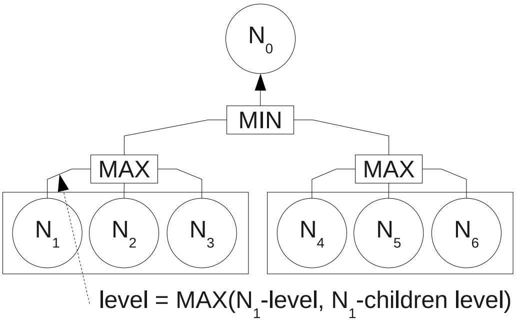


**Fig. 7.** Forest-based non-local pruning: after the propagation, each value node contains only lists of children with minimal rule level

## 4 Evaluation

The contribution of the new pruning mechanism has been evaluated using a test suite that has been developed for `Synt` [8]. The test suite uses the Brno Phrasal Treebank (BPT), a corpus of currently 5,599 Czech sentences and their phrasal syntactic trees, and performs several measurements such as tree-to-tree similarity using the leaf-ancestor assessment (LAA) [9] algorithm or basic analyses statistics including number of output syntactic trees.

From all the 5,599 sentences, 3,779 have been processed with unambiguous and 1,820 with ambiguous morphological annotation (plain text annotated by the Czech morphological analyser `ajka` [10,11]). In Table 1 an overall comparison is given for the whole set of sentences showing the difference in parser performance before and after

**Table 1.** Evaluation of the forest-based non-local pruning on the Brno Phrasal Treebank. LAA Best value refers to the best achieved LAA among the first 100 trees, LAA First is the LAA of the first tree.

| value | before | after |
|---|---|---|
| # of sentences | 5,599 | |
| # of not accepted sent. | 202 | 116 |
| **median trees count** | **80** | **16** |
| **average trees count** | **2,175,562.7** | **41,388.3** |
| LAA Best | 0.8973 | 0.9016 |
| position of LAA Best | 26 | 17 |
| LAA First | 0.8578 | 0.8649 |
| Time elapsed [1] | 0:28:23 | 0:21:54 |
| Time per sentence [1] | 0.28 s | 0.21 s |

[1] Note that this is all-included time that entails analysing each sentence twice, computing LAA and all other statistics. Raw time of analysing each sentence once is ca. 25 % of this one.

implementing the forest-based pruning method. A significant drop-off in the ambiguity can be seen as the average number of syntactic trees has been reduced by two orders of magnitude and the median dropped to quarter of its former value – all of this without harming parsers accuracy that has been actually even slightly improved. A noticeable speedup of the analysis has been achieved as well, conforming to the reduced size of the forest of values.

In Table 2 the results are provided separately for sentences with ambiguous and unambiguous morphological annotation. We can see that in the unambiguous case, the newly implemented methods lead to pruning up to dozens of trees. This is extremely important since it enables to perform an exhaustive search of the remaining trees and to use deep and possibly time-consuming methods to rerank such a small set of trees.

**Table 2.** Evaluation of the forest-based non-local pruning separated for sentences with ambiguous and unambiguous morphological annotation

| | **unambiguous** | | **ambiguous** | |
|---|---|---|---|---|
| **value** | **before** | **after** | **before** | **after** |
| # of sentences | 3,779 | | 1,820 | |
| # of not accepted sent. | 114 | 73 | 88 | 43 |
| **median trees count** | **24** | **6** | **6144** | **480** |
| **average trees count** | **12,601.5** | **144.6** | **6,768,409.0** | **127,525.9** |
| LAA Best | 0.9128 | 0.9131 | 0.8643 | 0.8776 |
| position of LAA Best | 19 | 8 | 43 | 36 |
| LAA First | 0.8784 | 0.8833 | 0.8141 | 0.8263 |

## 5 Conclusions

The new pruning method described in this paper effectively reduces the number of resulting syntactic trees to the extent where we can afford using tree-based reranking techniques to continue improving the precision of the parser. It also allows further development of the metagrammar used by `Synt` in the form of competing CFG rules and extend the coverage of the parser without harming its accuracy by increasing the overgeneration of the grammar.

The measured results also confirm the fact that the ambiguity of the morphological annotation of the input still plays a key role in the ambiguity of the parsing itself – while the above mentioned results are satisfiable for unambiguous input, the ambiguous case still remains to be an issue.

In the future we would like to evaluate the contribution on application-driven tests such as extraction of phrases [12], building word sketches from a corpus [13], exploiting into dictionary tools [14] or semantic role labeling [15].

**Acknowledgements.** This work has been partly supported by the Ministry of the Interior of CR within the project VF20102014003, by EC FP7 project ICT-248307 and by the Czech Science Foundation under the project P401/10/0792.

## References

1. Zeman, D., Žabokrtský, Z.: Improving Parsing Accuracy by Combining Diverse Dependency Parsers. In: Proceedings of the 9th International Workshop on Parsing Technologies (2005)
2. Kadlec, V., Horák, A.: New Meta-grammar Constructs in Czech Language Parser Synt. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 85–92. Springer, Heidelberg (2005)
3. Horák, A., Holan, T., Kadlec, V., Kovář, V.: Dependency and Phrasal Parsers of the Czech Language: A Comparison. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 76–84. Springer, Heidelberg (2007)
4. Kovář, V., Horák, A., Kadlec, V.: New Methods for Pruning and Ordering of Syntax Parsing Trees. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 125–131. Springer, Heidelberg (2008)
5. Kadlec, V.: Syntactic analysis of natural languages based on context-free grammar backbone. PhD thesis, Faculty of Informatics, Masaryk University, Brno (2007)
6. Sikkel, K.: Parsing Schemata – A Framework for Specification and Analysis of Parsing Algorithms. Springer, Heidelberg (1997)
7. Jakubíček, M., Horák, A., Kovář, V.: Mining Phrases from Syntactic Analysis. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 124–130. Springer, Heidelberg (2009)
8. Kovář, V., Jakubíček, M.: Test Suite for the Czech Parser Synt. In: Proceedings of Recent Advances in Slavonic Natural Language Processing 2008, Brno, pp. 63–70 (2008)
9. Sampson, G., Babarczy, A.: A Test of the Leaf-Ancestor Metric for Parse Accuracy. Natural Language Engineering 9(04), 365–380 (2003)
10. Šmerk, P.: Unsupervised learning of rules for morphological disambiguation. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 211–216. Springer, Heidelberg (2004)
11. Pala, K., Rychlý, P., Šmerk, P.: Morphological Analysis of Law Texts. In: Proceedings of Recent Advances in Slavonic Natural Language Processing 2007, pp. 21–26 (2007)
12. Grác, M., Jakubíček, M., Kovář, V.: Through low-cost annotation to reliable parsing evaluation. In: 24th Pacific Asia Conference on Language, Information and Computation (2010)
13. Rychlý, P.: Manatee/Bonito - A Modular Corpus Manager. In: Proceedings of Recent Advances in Slavonic Natural Language Processing 2007, Brno, Masaryk University (2007)
14. Horák, A., Pala, K., Rambousek, A.: The Global WordNet Grid Software Design. In: Proceedings of the Fourth Global WordNet Conference, University of Szegéd, pp. 194–199 (2008)
15. Nevěřilová, Z.: Semantic Role Patterns and Verb Classes in Verb Valency Lexicon. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 150–156. Springer, Heidelberg (2010)

# Efficiency of Speech Alignment for Semi-automated Subtitling in Dutch

Patrick Wambacq and Kris Demuynck

ESAT/PSI-Speech, Katholieke Universiteit Leuven, Belgium
{wambacq,krisdm}@esat.kuleuven.be

**Abstract.** This paper describes the use of speech alignment to aid in the process of subtitling Dutch TV programs. The recognizer aligns the audio stream with an existing transcript. The goal is therefore not to transcribe but to generate the correct timing of every word. The system performs subtasks such as audio segmentation, transcript preprocessing, alignment and subtitle compression. The result is not perfect but good enough to gain efficiency when used by a professional subtitler as a starting point to refine and finalize the subtitles. In our tests, considerable time savings of 47 to 53% on average are obtained, such that the generation of subtitles for a 1 hour program, is lowered from between 4 and 7 hours to between 2.5 and 4 hours. This is all the more important in the context of an increased pressure from user groups on governments and broadcasters to reach 100% subtitled TV programs.

## 1 Introduction

Organizations of the deaf and hard of hearing are since long pushing broadcasters and governments to increase the amount of subtitled television programs. This asks for large investments in new technologies and personnel. One of the technologies that can come to aid is speech recognition. In principle, speech recognition can be used in several ways to produce subtitles:

- Generate the transcript and use this as the basis for subtitles. Depending on the type of TV program (e.g. documentary vs. discussions on politics) this works rather well or not at all. Although possible for some tasks, on average word error rate (WER) is too high and due to the numerous required manual corrections in the subsequent post-editing step, no time is saved in that case.
- A well trained speaker re-speaks the audio and an automatic speech recognition (ASR) system adapted to his voice produces a good quality transcript that is already more or less time-synchronized to the soundtrack. For best quality it should however be manually checked (this time requiring much less time given the quality of the source transcript), and perfectly time-aligned to the soundtrack.
- When transcripts are available (either from a re-speaking step as described above or from the program production) they can be aligned to the audio using a speech recognizer. The result is manually checked in a post-editing step but time savings are considerable.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 123–130, 2011.

Several efforts to use speech recognizers for subtitle generation have been reported, e.g. [1] transcribes broadcast news and [2] uses both the re-speaking approach and transcription. However the context is too different to be able to compare results. In a project called NEON (“Nederlandstalige Ondertiteling”, Dutch for “Dutch Subtitling”)[1] technology providers and broadcasters have teamed to evaluate the third approach. The results of this evaluation are presented here.

The organization of the paper is as follows: first the segmentation, transcript preprocessing and alignment subtasks are described. Then the complete system is presented. The results of the evaluation of the system are discussed and finally conclusions are given.

## 2 Segmentation of the Audio Stream

This stream-based subsystem detects long intervals of non-speech that can be discarded in further processing. It produces the following segmentations with low delay and computational effort:

- speech/non-speech: feed the aligner with speech only segments to lower its error rate and processing time;
- male/female: allow to select an appropriate male or female acoustic model;
- speaker clustering: used for speaker adaptation through speaker specific vocal tract length normalization (VTLN) and spectral mean normalization.

The speech/non-speech segmentation uses gaussian mixture models (GMM’s) to distinguish several categories of audio events (speech, music, speech+music, speech+other). The speaker segmentation and clustering is based on a Bayesian Information Criterion (BIC). The details of the system can be found in [3].

## 3 Preprocessing of the Transcripts

Next to the audio stream, the second input to the aligner is the available transcript which also contains any or all of the following metadata: speaker identities, timing information, stage directions, description of music cues, etc. This meta information is not fed to the aligner, but is kept aside since it can improve the quality of the generated subtitles. In a merging step it augments the alignment results with extra information. Therefore the transcripts are split into the true transcript (which is tokenized) and the metadata.

## 4 Speech Alignment

The alignment step takes at its input both the soundtrack and the tokenized transcript and generates a time aligned output, i.e. every word receives exact begin and end times,

[1] The NEON project is carried out within the STEVIN program which is funded by the Dutch and Flemish Governments. (http://www.stevin-tst.org)

needed to show the subtitle at the right moment. To this end the SPRAAK system [4] is used in recognition mode (and not in alignment mode) with a restricted finite state grammar (FSG) as explained below.

In the **preprocessing stage** simple speaker adaptation is performed through speaker specific spectral mean normalization and VTLN.

The **language model** (LM) consists of a FSG that is built from the input transcript. First, every sentence in the transcript is numbered and assigned to a time window through a linear interpolation rule that takes into account the lengths of the sentence, of the transcript and of the soundtrack. For every segment that is labeled as speech, a set of candidate sentences is constructed. This set contains the sentence that is closest in time to the segment, and a number of previous and following sentences. The set is then used to construct a small FSG that serves as the language model for the alignment. This means that for every speech segment, a dedicated LM is constructed. The set is expanded or contracted and time shifted as the aligner steps through the sequence of segments following some heuristics. This keeps a small number of possibilities to choose from by the aligner while at the same time allows to cope with deviations between transcript and audio. Two types of deviations can occur.

Firstly, there can be extra sentences in the transcript: this occurs typically when the direction decides last minute changes to what is aired (usually to shorten an item). This type of deviation is taken care of by the above heuristic as long as the skips are not too large. In the presence of very large skips, the aligner derails and the user has to restart it from the point where it goes wrong by adjusting the set of candidate sentences. This was observed now and then but given the fast processing time this does not pose many problems (and only the remaining part needs to be aligned again).

Secondly there may be non transcribed audio: obviously whatever the aligner tries to map onto it, it will not be correct. The aligner gets back on track if the set of candidate sentences still contains the correct sentence for the subsequent piece of transcribed audio (i.e. when the non transcribed audio is not too long). In the future we will add a fallback to full recognition mode with a general language model to try to remedy this.

The generated **lexicon** contains all words of the transcript. It is created by an updated version of the system described in [6]. The core lexicon is Fonilex [7] which provides multiple phonetic transcriptions for 170k common Dutch words. Based on a simple classification (initial capital, all capitals, etc.), the remaining words are sent to one or more of the following modules:

- An inflection, derivation and compounding module which finds possible decompositions and merges the phonetic transcriptions of the composing parts using the appropriate assimilation rules.
- A module that handles letter/digits words such as acronyms.
- A grapheme to phoneme (g2p) converter. The g2p system was trained on the Fonilex lexicon and hence produces pronunciations in line with that of standard Dutch words. When handling proper nouns, some rules are first applied to convert the archaic spelling conventions commonly used in Dutch names to a more modern form. Nevertheless, the automatic transcription of proper nouns (especially from foreign origin) remains problematic.

Fonilex also provides rules to generate the alternative pronunciation variants of a word. An extended version of this rule set was used to generate all likely pronunciation variants which resulted in a median of 3.8 pronunciations per word or 1.13 variants per phone in the canonical transcriptions of the words.

The **acoustic model** (AM) is taken from a Flemish Broadcast News transcription task [5]. In summary, this is a triphone HMM with state emissions modeled by GMMs with globally tied gaussians (4k states and a pool of 50k gaussians), based on MIDA features (mutual information based discriminant analysis) and using 49 three-state cross-word triphones and one single state triphone (short schwa). Since the transcript is known, the search space is very restricted and hence the AM is not very critical to the performance.

## 5 Overview of the Complete System

A block diagram of the complete system is shown in Fig. 1. The different subtasks (blocks in the figure) are controlled by a PRAAT script. The automatic sentence compression block is not discussed here because it has nothing to do with the speech alignment. Readers who want to know more about it are referred to [8]. We have chosen for the PRAAT software [9] to control the process because it provides a simple way to present the audio signal and several tiers are available that can be tailored to many tasks. Although not optimal, this was certainly a quick and good way to demonstrate the power of the system and to let the broadcasters evaluate its potential. In a real subtitling application, a specialized GUI should be developed or the subtitle aligner should be integrated in an existing software environment for subtitling. In our case, different tiers indicate the result from several steps: speech/non-speech detection (tier 2 in Fig. 2), word alignment (tier 1), subtitle alignment (tier 3). Tier 4 shows the set of candidate sentences that the aligner can choose from for the current segment, as discussed in Sect. 4. The user can change the contents of the tiers if required (to correct a wrong segmentation, to put the aligner back on track after alignment errors, to eliminate non-transcribed foreign speech, etc.).

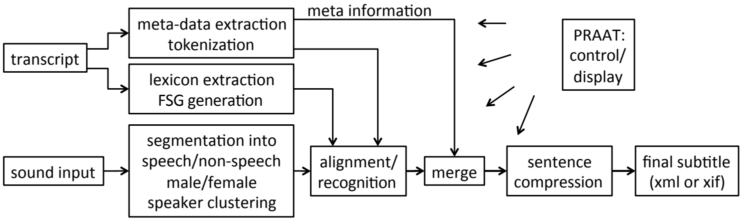
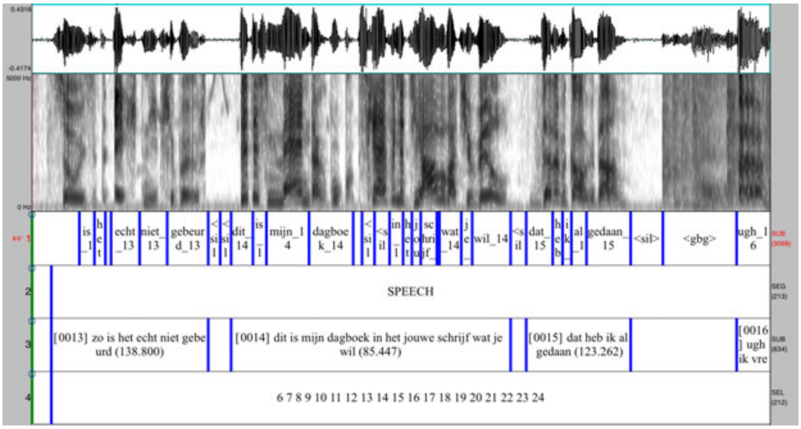


**Fig. 1.** Block diagram of the system

The system's speed is shown in the table below (average figures for 20 minutes of broadcast audio, measured on a 3 GHz Intel core2duo machine (using only one core however) with 2 GBytes of memory, running WinXP SP3).


0.4316
-0.4174
5000 Hz
0 Hz
1
is _1 h e t echt _13 niet_ 13 gebeur d_13 < si l < si l dit_ 14 is _1 mijn_1 4 dagboe k_14 < si l <s il in _1 h e t j o u sc hri jf_ wat _14 j e wil_14 <s il dat_ 15 h e b i k al _1 gedaan_ 15 <sil> <gbg> ugh_1 6
SUB (3068)
2
SPEECH
SEG (213)
3
SUB (634)
4
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
SEL (212)

[0013] zo is het echt niet gebeurd (138.800)

[0014] dit is mijn dagboek in het jouwe schrijf wat je wil (85.447)

[0015] dat heb ik al gedaan (123.262)

[0016] ugh ik vre

**Fig. 2.** PRAAT screen shot with the tiers indicating results of different steps

| | |
|---|---|
| Soundtrack extraction from video | 28 sec |
| Transcript preprocessing | 1 sec |
| Segmentation and clustering | 34 sec |
| Lexicon generation | 4 sec |
| Alignment | 434 sec |
| Subtitle compression | ≈ 1 sec/sentence |
| Total | ≈ 550 sec |

On average, the system is a little more than two times faster than real-time. This does not include any human intervention. Time gains that include human post-editing are discussed in Sect. 6.

## 6 System Evaluation

A prototype of the system was evaluated by broadcasters in Flanders (VRT) and the Netherlands (NPO) on a large variety of programs: documentaries, soaps, animation, human interest, programs for children, action series, church service, etc. with a varying degree of intermixed foreign speech parts and voice-overs that were classified as speech by the segmentation step.

### 6.1 Qualitative Evaluation

The general impression of the users was very positive. At the beginning, the learning curve (especially the use of the PRAAT controls) was a bit steep, but after a while the application ran very smooth and proved to be very robust. The software saves lots of time compared to the manual process and the quality of the result was much better than expected.

Sentence compression was rarely used (although of good quality): it was either not required or solved differently, because either the alignment was done with already condensed transcripts (obtained through e.g. re-speaking) or the human editor who conducted the post-editing took care of it manually.

As expected, the PRAAT interface was deemed not optimal, since it was chosen for rapid prototyping, only demonstrating the potential of speech alignment.

### 6.2 Quantitative Evaluation

A quantitative evaluation was also pursued. Since this is an alignment task, word error rate is not the right metric. The Levenshtein distance between the aligned subtitles and some ground truth could be calculated but this does not consider timing. Also there is no real ground truth since there is not such a thing as a single perfect subtitle: every human subtitler produces slightly different results. Moreover, a Levenshtein distance does not indicate how much time saving the approach would deliver (although we can suspect that there is some correlation between both). Another measure that can be calculated is a histogram of the deviations between the correct timing and generated timing of the subtitles. Subtitlers tell us that segment boundaries should lie within 200 msec from the exact times. A previous experiment on English subtitles in another project, demonstrated that more than 97% of the subtitles met this criterion ([10]). This calculation was not undertaken for the current project but given the improvements in acoustic modelling and alignment implemented in NEON, similar or better results are expected. Also this measure does not tell anything about time savings. Fig. 3 shows a plot of timing errors (for a test on an english program in another project, [10]), clearly indicating that these are not frequent.

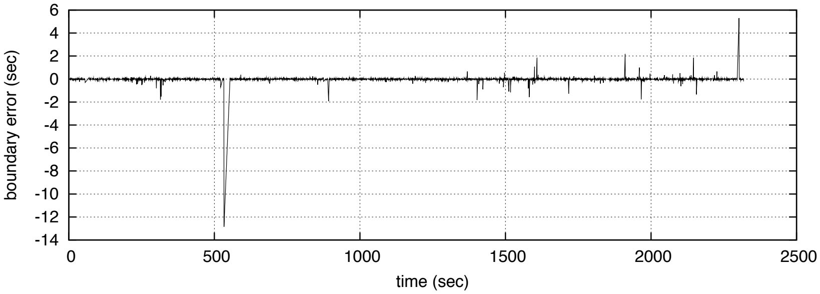


**Fig. 3.** Alignment timing errors

The only useful evaluation metric in our view is the time saved. This was measured by professional subtitlers on a mix of TV programs, with and without the aid of our system (called manual and NEON further on). Manual subtitle generation as well as post-editing of the automatically generated subtitles are performed with existing commercial software (Swift by Softel for VRT and WinCAPS by SysMedia for NPO) whose functionality is not addressed here. The same program was never subtitled by the same person using both approaches to avoid a possible influence of the first result on the second one.

VRT conducted a test where the same person was given different programs from the same series (e.g. two consecutive episodes) for the manual and NEON tests. After producing the subtitle, a final check was performed (running along with the TV program, so its duration equals the audio length). VRT confirmed after the evaluation that this extra step is not really required. The results are shown in the table below, indicating a 20% efficiency gain. The ”NEON time” shows the time that the subtitler needs to control and use the NEON aligner with PRAAT; it does not correspond to the required CPU time. The column ”manual time” shows the time required to produce the initial subtitle starting from the NEON result (or from scratch). When the final check that was not deemed necessary is removed, the time gain amounts to 53% for VRT.

| VRT (total audio length of the test: 725 min); times below in mins | | | | | |
|---|---|---|---|---|---|
| # of programs | NEON time | manual time | final check | total | RT factor |
| 15 (manual) | 0 | 3130 | 520 | 3650 | 7.02 |
| 7 (NEON) | 365 | 585 | 205 | 1155 | 5.63 |
| efficiency gain = 100*(7.02-5.63)/7.02=20% | | | | | |
| efficiency gain without final check = 53% | | | | | |

NPO conducted a test with 9 programs. The manual and NEON tests were performed by different persons. The results in the table below show a 47% efficiency gain. Here the column ”total time NEON” includes both the time needed to control the automatic alignment and the manual refining time.

| NPO (total audio length of the test: 149 min); times below in mins | | |
|---|---|---|
| # of programs | total time manual | total time NEON |
| 9 | 1390 | 740 |
| efficiency gain = 100*(1390-740)/1390=47% | | |

The difference in gain between the two broadcasters is attributed to differences in experience that the subtitlers have gained with the application, to different procedures when subtitling and to differences in the TV programs.

The broadcasters made several suggestions to further increase the time savings, mainly concerning the user interface (NPO estimates a further potential 45% speedup from a limited set of changes to the GUI). They also found some errors in the PRAAT scripts that control the alignment, that led to some extra alignment errors. Fixing these errors therefore would increase the time savings.

## 7 Conclusions

In this paper we have reviewed the use of speech alignment in the generation of subtitles for TV programs. Although the aligned subtitles are not always correct and human intervention in a post-editing step is still required, the time savings are considerable (47 to 53% on average in our tests). The limitation of our approach is that a transcript is needed. This can however be produced by the re-speaking approach by a trained speaker, after which the generated transcription can be used as any other script in our approach. This method would also save time since re-speaking happens in real-time and the generated transcription would be of high quality requiring only minor post-editing on the subtitles that are based on it.

We regard the obtained time savings as minimum values. By fixing some errors, optimizing the alignment, providing automatic language detection and providing a user interface targeted at semi-automatic subtitling, the gains will increase further. We will also add fallback to full recognition mode in a next version. Our experiments described here are only the first steps towards a successful ASR aided subtitling system for Dutch.

## References

1. Meinedo, H., Viveiros, M., da Silva Neto, J.P.: Evaluation of a Live Broadcast News Subtitling System for Portuguese. In: Proc. Interspeech 2008, Brisbane, Australia, pp. 508–511 (September 2008)
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. Vandecatseye, A., Martens, J.-P.: A Fast, Accurate and Stream-Based Speaker Segmentation and Clustering Algorithm. In: Proceedings of the 8th European Conference on Speech Communication and Technology, Eurospeech 2003, Geneva, Switzerland, vol. 2, pp. 941–944 (September 2003)
4. Demuynck, K., Roelens, J., Van Compernolle, D., Wambacq, P.: SPRAAK: An Open Source SPeech Recognition and Automatic Annotation Kit. In: Proc. Interspeech 2008, Brisbane, Australia, p. 495 (September 2008)
5. Demuynck, K., Puurula, A., Van Compernolle, D., Wambacq, P.: The ESAT 2008 System for N-Best Dutch Speech Recognition Benchmark. In: Proc. IEEE ASRU Workshop, Merano, Italy, pp. 339–343 (December 2009)
6. Demuynck, K., Laureys, T., Wambacq, P., Van Compernolle, D.: Automatic phonemic labeling and segmentation of spoken Dutch. In: Proc. LREC-2004, Lisbon, Portugal, pp. 61–64 (May 2004)
7. Mertens, P., Vercammen, F.: ”FONILEX manual”, K.U.Leuven – CCL Technical report (1998), http://bach.arts.kuleuven.be/fonilex
8. Daelemans, W., Höthker, A., Tjong Kim Sang, E.: Automatic sentence simplification for subtitling in Dutch and English. In: Proc. LREC 2004, Lisbon, Portugal, pp. 1045–1048 (May 2004)
9. Boersma, P., Weenink, D.: Praat: doing phonetics by computer (Version 5.1.05) (Computer program), from http://www.praat.org/ (retrieved May 1, 2009)
10. Wambacq, P., Vanroose, P., Yang, X., Duchateau, J., Van Uytsel, D.H.: Speech Recognition for Subtitling Purposes. In: Proc. 5th Intl. Conf. Languages & The Media, Berlin, Germany, p. 46 (November 2004) (Abstract)

# Evaluation of Hands-Free Large Vocabulary Continuous Speech Recognition by Blind Dereverberation Based on Spectral Subtraction by Multi-channel LMS Algorithm

Longbiao Wang, Kyohei Odani, and Atsuhiko Kai

Department of Systems Engineering, Shizuoka University, Japan
3-5-1 Johoku, Naka-ku, Hamamatsu 432-8561, Japan
{wang,odani,kai}@spa.sys.eng.shizuoka.ac.jp

**Abstract.** Previously, Wang et al. [1] proposed a blind dereverberation method based on spectral subtraction using a multi-channel least mean squares (MCLMS) algorithm for distant-talking speech recognition. Preliminary experiments showed that this method is effective for isolated word recognition in a reverberant environment. However, robustness and effect factors of the dereverberation method based on spectral subtraction were not investigated. In this paper, we analyze the effect factors of compensation parameter estimation for the dereverberation method based on spectral subtraction, such as the number of channels (the number of microphones), the length of reverberation to be suppressed, and the length of the utterance used for parameter estimation, and evaluate these on large vocabulary continuous speech recognition (LVCSR). We conducted speech recognition experiments on a distorted speech signal simulated by convolving multi-channel impulse responses with clean speech. The proposed method with beamforming achieves a relative word error reduction rate of 19.2% relative to conventional cepstral mean normalization with beamforming for LVCSR. The experimental results also show that our proposed method is robust in a variety of reverberant environments for both isolated and continuous speech recognition and under various parameter estimation conditions.

## 1 Introduction

In a distant-talking environment, channel distortion drastically degrades speech recognition performance due to a mismatch between the training and testing environments. Compensating an input feature is the main method for reducing the mismatch. Cepstral mean normalization (CMN) has been especially employed as a simple and effective way of normalizing the cepstral feature to reduce channel distortion [2]. However, the impulse response of reverberation in a distant-talking environment usually has a much longer tail than the window length of the short-term spectral analysis. Therefore, conventional CMN is not totally effective under these conditions. Several studies have focused on mitigating this problem [3,4]. A reverberation compensation method for speaker recognition using spectral subtraction in which late reverberation is treated as additive noise was proposed in [4]. However, the drawback of this approach is that the optimum parameters for spectral subtraction are empirically estimated from a development dataset and late reverberation cannot be subtracted correctly as it is not modeled precisely.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 131–138, 2011.

In a previous work [1], Wang et al. proposed a robust distant-talking speech recognition method based on power spectral subtraction (SS) employing the adaptive multichannel least mean squares (MCLMS) algorithm. We treated the late reverberation as additive noise, and a noise reduction technique based on power SS was proposed to estimate the power spectrum of clean speech using an estimated power spectrum of the impulse response. To estimate the power spectra of the impulse responses, we extended the variable step-size unconstrained MCLMS (VSS-UMCLMS) algorithm for identifying impulse responses in a time domain [5,6] to a frequency domain. The early reverberation was normalized by Cepstral Mean Normalization (CMN). By combining the proposed method with beamforming, a relative error reduction rate of 24.5% compared to the conventional CMN with beamforming was achieved on an isolated word recognition task.

In this paper, we investigate the robustness of the method proposed in [1] under various reverberant conditions for both isolated word recognition and large vocabulary continuous speech recognition (LVCSR). We also analyze the effect factors (numbers of reverberation windows and channels, length of utterance, and the distance between sound source and microphone) of compensation parameter estimation for dereverberation based on spectral subtraction.

## 2 Dereverberation Based on Power Spectral Subtraction

If speech $s[t]$ is corrupted by convolutional noise $h[t]$ and additive noise $n[t]$, the observed speech $x[t]$ becomes

$$x[t] = h[t] * s[t] + n[t]. \tag{1}$$

In this paper, additive noise is ignored for simplification, so Eq. (1) becomes $x[t] = h[t] * s[t]$.

If the length of the impulse response is much smaller than the size $T$ of the analysis window used for short-time Fourier transform (STFT), the STFT of the distorted speech equals that of the clean speech multiplied by the STFT of the impulse response $h[t]$. However, if the length of the impulse response is much greater than the analysis window size, the STFT of the distorted speech is usually approximated by

$$\begin{aligned} X(f,\omega) &\approx S(f,\omega) * H(\omega) \\ &= S(f,\omega)H(0,\omega) + \sum_{d=1}^{D-1} S(f-d,\omega)H(d,\omega), \end{aligned} \tag{2}$$

where $f$ is the frame index, $H(\omega)$ is the STFT of the impulse response, $S(f,\omega)$ is the STFT of clean speech $s$ and $H(d,\omega)$ denotes the part of $H(\omega)$ corresponding to the frame delay $d$. That is, with a long impulse response, the channel distortion is no longer of a multiplicative nature in a linear spectral domain, but is rather convolutional [3].

We compensate the early reverberation by subtracting the cepstral mean of the utterance. Assuming that phases of different frames is noncorrelated for simplification, the power spectrum of Eq. (2) can be approximated as [1]

$$|\tilde{X}(f,\omega)|^2 \approx |\tilde{S}(f,\omega)|^2 + \frac{\sum_{d=1}^{D-1}\{|\tilde{S}(f-d,\omega)|^2|H(d,\omega)|^2\}}{|H(0,\omega)|^2}, \tag{3}$$

where $|\tilde{S}(f,\omega)|^2 = \frac{|S(f,\omega)|^2}{|\bar{S}(f,\omega)|^2}$, and $\bar{S}(f,\omega)$ is the mean vector of $S(f,\omega)$. The power spectrum of clean speech $|\hat{S}(f,\omega)|^2$ can be estimated as

$$|\hat{S}(f,\omega)|^2 \approx max\{|\tilde{X}(f,\omega)|^2 -$$

$$\alpha \cdot \frac{\sum_{d=1}^{D-1}\{|\tilde{S}(f-d,\omega)|^2|H(d,\omega)|^2\}}{|H(0,\omega)|^2}, \beta \cdot |X(f,\omega)|^2\}. \tag{4}$$

where $\alpha$ is the noise over estimation factor, $\beta$ is the spectral floor parameter to avoid negative or under flow values, and $H(d,\omega), d = 0,1...D-1$ is the STFT of the impulse response, which can either be calculated from a known impulse response or be blindly estimated. $D$ is the number of reverberation windows. In this paper, the spectrum of the impulse response for the spectral subtraction is blindly estimated using the method described in Section 3.

## 3 Compensation Parameter Estimation for Spectral Subtraction by Multi-channel LMS Algorithm

In [6], an adaptive multi-channel LMS algorithm for blind Single-Input Multiple-Output (SIMO) system identification was proposed.

In the absence of additive noise, we can take advantage of the fact that

$$x_i * h_j = s * h_i * h_j = x_j * h_i,\ i,j = 1,2,\cdots,N, i \neq j, \tag{5}$$

and have the following relations at time $t$:

$$\mathbf{x}_n(t) = [x_n(t)\ \ x_n(t-1)\ \ ...\ \ x_n(t-L+1)]^T, \tag{6}$$

$$\mathbf{h}_n(t) = [h_n(t,0)\ \ \ h_n(t,1)\ \ \ ...\ \ \ h_n(t,L-1)]^T, \tag{7}$$

$$n = 1,2,...,N,$$

where $n$ is the channel index, $\mathbf{x}_n(t)$ is the speech signal received at time $t$, $\mathbf{h}_n(t)$ is the impulse response at time $t$, $h_n(t,l)$ is the *l-th* tap of the impulse response at time $t$, and $L$ is the number of taps of the impulse response.

An estimated error vector at time $t$ is expressed as

$$e_{ij}(t+1) = \mathbf{x}_i^T(t+1)\mathbf{h}_j(t) - \mathbf{x}_j^T(t+1)\mathbf{h}_i(t), \tag{8}$$

$$i,j = 1,2,...,N, i \neq j.$$

This error can be used to define a cost function at time $t$

$$\mathbf{J}(t+1) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} e_{ij}^2(t+1). \tag{9}$$

**Table 1.** Detailed recording conditions for impulse response measurement. "angle": recording direction between microphone and loudspeaker; "RT60 (second)": reverberation time in room; "S": small; "L": large.

| array no | array type | room | angle | RT60 |
|---|---|---|---|---|
| 1 | linear | tatami-floored room (S) | 120° | 0.47 |
| 2 | circle | tatami-floored room (S) | 120° | 0.47 |
| 3 | circle | tatami-floored room (L) | 90° | 0.60 |
| 4 | circle | tatami-floored room (L) | 130° | 0.60 |
| 5 | linear | Conference room | 50° | 0.78 |
| 6 | linear | echo room (panel) | 70° | 1.30 |
| 7 | linear | echo room (panel) | 150° | 0.30 |
| 8 | circle | echo room (cylinder) | 30° | 0.38 |

By minimizing the cost function $J$ of Eq.(9), the impulse response can be blindly derived. We extended this VSS-UMCLMS algorithm [5,6], which identifies the multi-channel impulse responses, for processing in a frequency domain with SS applied in combination [1].

## 4 Experiments

### 4.1 Experimental Setup

Multi-channel distorted speech signals simulated by convolving multi-channel impulse responses with clean speech were used to evaluate our proposed algorithm. Eight kinds of multi-channel impulse responses measured in various acoustical reverberant environments were selected from the Real World Computing Partnership sound scene database [7]. A four-channel circular or linear microphone array was taken from a microphone array with 30 channels. The four-channel circular type microphone array had a diameter of 30 $cm$, and the four microphones were located at equal 90° intervals. The four microphones of the linear microphone array were located at 11.32 $cm$ intervals. Impulse responses were measured at several positions 2 $m$ from the microphone array. The sampling frequency was 48 $kHz$. Table 1 lists the conditions for the eight recordings using a four-channel microphone array. The Japanese Newspaper Article Sentences (JNAS) corpus [8] was used as clean speech. 100 utterances from the JNAS database convolved with the multi-channel impulse responses shown in Table 1 were used as test data. The average duration of all utterances was about 5.8 $s$.

Table 2 gives the conditions for speech recognition. The acoustic models were trained with the Acoustic Society of Japan (ASJ) speech databases of phonetically balanced sentences (ASJ-PB) and the JNAS. In total, around 20K sentences (clean speech) uttered by 132 speakers were used for each gender. Table 3 gives the conditions for spectral subtraction based dereverberation. The parameters shown in Table 3 were determined empirically. An illustration of the analysis window is shown in Fig. 1. For the proposed dereverberation method based on spectral subtraction, the previous clean power spectra estimated with a skip window were used to estimate the current clean

**Table 2.** Conditions for speech recognition

| | |
|---|---|
| sampling frequency | 16 kHz |
| frame length | 25 ms |
| frame shift | 10 ms |
| acoustic model | 3 output probability left-to-right triphone HMMs |
| feature space | 25 dimensions with CMN (12MFCCs+$\Delta$+$\Delta$power) |

**Table 3.** Conditions for spectral subtraction based dereverberation

| | |
|---|---|
| analysis window | Hamming |
| window length | 32 ms |
| window shift | 16 ms |
| number of reverberant windows $D$ | 6<br>(192 ms) |
| noise overestimation factor $\alpha$ | 1.0 |
| spectral floor parameter $\beta$ | 0.15 |

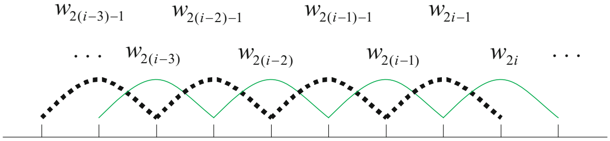


**Fig. 1.** Illustration of the analysis window for spectral subtraction

power spectrum since the frame shift was half the frame length in this study. The spectrum of the impulse response $H(d, \omega)$ was estimated for each utterance to be recognized. The word accuracy rate for LVCSR with clean speech was 92.6%.

### 4.2 Experimental Results on LVCSR

In this paper, four microphones were used to estimate the spectrum of the impulse responses. Delay-and-sum beamforming (BF) was performed on the 4-channel dereverberant speech signals. For the proposed method, each speech channel was compensated by the corresponding estimated impulse response. Preliminary experimental results for isolated word recognition showed that the proposed method significantly improved speech recognition performance compared with traditional CMN with beamforming [1].

In this paper, we also evaluated our proposed method on LVCSR with the experimental results shown in Fig. 2. Naturally, the speech recognition rate deteriorated as the reverberation time increased. Using the proposed method, the reduction in the speech recognition rate was smaller than in conventional CMN, especially for impulse responses with a long reverberation time. The proposed method achieved a relative word recognition error reduction rate of 19.2% relative to CMN with delay-and-sum beamforming. We also conducted an LVCSR experiment with dereverberation based on spectral subtraction under different reverberant conditions, with the reverberation time between 0.25 $s$ and 0.75 $s$ and the distance between microphone and sound source 0.5 $m$. A similar trend to the above results was observed. Details of the experimental setup and the results are not presented due to space limitations. Therefore, our proposed

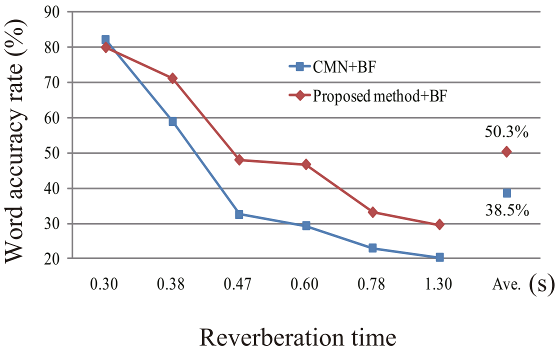


**Fig. 2.** Word accuracy rate for LVCSR

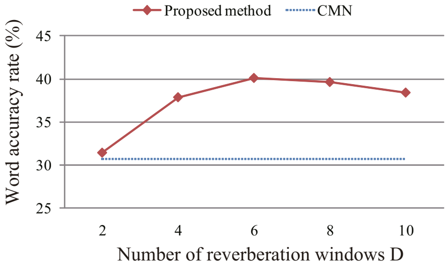


**Fig. 3.** Effect of the number of reverberation windows on speech recognition

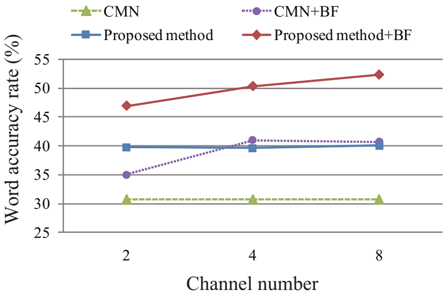


**Fig. 4.** Effect of the number of channels on speech recognition

method is robust to various reverberant conditions for both isolated word recognition and LVCSR. The reason is that our proposed method can compensate for late reverberation through spectral subtraction using an estimated power spectrum of the impulse response.

### 4.3 Effect Factor Analysis of Compensation Parameter Estimation

In this section, we analyze the effect factors (number of reverberation windows, number of channels, and length of utterance) of compensation parameter estimation for the dereverberation method based on spectral subtraction.

The effect of the number of reverberation windows on speech recognition is shown in Fig. 3. The optimal number of reverberation windows is 6. The speech recognition performance with the number of reverberation windows between 4 and 10 did not vary greatly and was significantly better than the baseline.

We also analyzed the influence of the number of channels on parameter estimation and delay-and-sum beamforming. Besides four channels, two and eight channels were also used to estimate the compensation parameter and perform beamforming. The results are shown in Fig. 4. The speech recognition performance of the proposed method

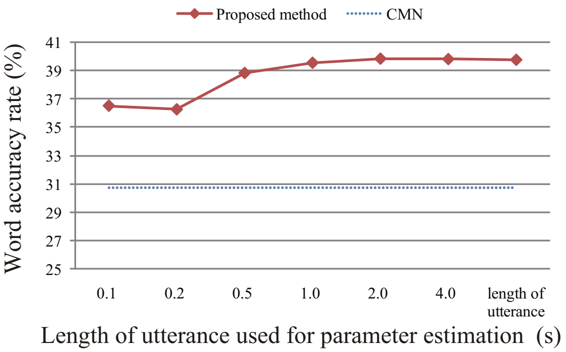


**Fig. 5.** Effect of length of utterance used for parameter estimation on speech recognition

without beamforming was hardly affected by the number of channels. That is, the compensation parameter estimation is robust to the number of channels. Combined with beamforming, the more channels that are used, the better is the speech recognition performance.

Thus far the whole utterance has been used to estimate the compensation parameter. The effect of the length of utterance used for parameter estimation was investigated, with the results shown in Fig. 5. The longer the length of utterance used, the better is the speech recognition performance. Deterioration in speech recognition was not experienced with the length of the utterance used for parameter estimation greater than $1\ s$. The speech recognition performance of the proposed method is better than the baseline even if only $0.1\ s$ of utterance is used to estimate the compensation parameter.

## 5 Conclusions and Future Work

In this paper, we proposed a blind reverberation reduction method based on spectral subtraction employing the multi-channel LMS algorithm for distant-talking large vocabulary continuous speech recognition. We treated late reverberation as additive noise, and a noise reduction technique based on spectral subtraction was proposed to estimate the clean power spectrum. The power spectrum of the impulse response was used to estimate the clean power spectrum. To estimate the power spectra of the impulse responses, we extended the MCLMS algorithm for identifying impulse responses in a time domain to a frequency domain. Our proposed algorithms were evaluated using distorted speech signals simulated by convolving multi-channel impulse responses with clean speech taken from the JNAS database. The proposed method achieves a 19.2% reduction in average error rate relative to that of conventional CMN. Our proposed method was also found to be robust to various reverberant environments for both isolated word recognition and LVCSR.

In this paper, we also investigated the effect factors (numbers of reverberation windows and channels, and length of utterance) for compensation parameter estimation. We reached the following conclusions: 1) the speech recognition performance with the number of reverberation windows between 4 and 10 did not vary greatly and was significantly better than the baseline; 2) the compensation parameter estimation was robust to the number of channels; and 3) degradation of speech recognition did not occur with the length of utterance used for parameter estimation longer than 1 $s$.

Additive noise has not been considered in this paper. In the future, we intend evaluating our proposed methods using real-world speech data simultaneously degraded by additive noise and convoluted noise.

**Acknowledgements.** This work has been supported in part by the NEDO Grant for Industrial Technology Research Program No. 07A12203a.

## References

1. Wang, L., Kitaoka, N., Nakagawa, S.: Distant-talking speech recognition based on spectral subtraction by multi-channel LMS algorithm. IEICE Trans. Information and Systems E94-D(3), 659–667 (2011)
2. Furui, S.: Cepstral analysis technique for automatic speaker verification. IEEE Trans. Acoust. Speech Signal Processing 29(2), 254–272 (1981)
3. Raut, C., Nishimoto, T., Sagayama, S.: Adaptation for long convolutional distortion by maximum likelihood based state filtering approach. In: Proc. of ICASSP-2006, vol. 1, pp. 1133–1136 (2006)
4. Jin, Q., Schultz, T., Waibel, A.: Far-field speaker recognition. IEEE Trans. ASLP 15(7), 2023–2032 (2007)
5. Huang, Y., Benesty, J., Chen, J.: Optimal step size of the adaptive multi-channel LMS algorithm for blind SIMO identification. IEEE Signal Processing Letters 12(3), 173–175 (2005)
6. Huang, Y., Benesty, J., Chen, J.: Acoustic MIMO Signal Processing. Springer, Heidelberg (2006)
7. Nakamura, S., Hiyane, K., Asano, F., Nishiura, T., Yamada, T.: Acoustical Sound Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition. In: Proc. of LREC 2000, pp. 965–968 (May 2000)
8. Itou, K., Yamamoto, M., Takeda, K., Takezawa, T., Matsuoka, T., Kobayashi, T., Shikano, K., Itahashi, S.: JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research. J. Acoust. Soc. Jpn (E) 20(3), 199–206 (1999)

# Fusion of Discriminative and Generative Scoring Criteria in GMM-Based Speaker Verification*

Boštjan Vesnicer[1], Jerneja Žganec Gros[2], and France Mihelič[2]

[1] Alpineon d.o.o., Ulica Iga Grudna 15, SI-1000 Ljubljana
{bostjan.vesnicer,jerneja.gros}@alpineon.com
[2] Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, SI-1000 Ljubljana
france.mihelic@fe.uni-lj.si

**Abstract.** The aim of this paper is to demonstrate the complementarity of different scoring methods used in speaker verification. To show that, we implemented two different scoring methods on top of the joint factor analysis model. The results on the telephone part of the NIST's SRE 2008 core condition show that significant increase in performance can be achieved by fusing likelihood ratio- and support vector machine-based scores.

**Keywords:** speaker verification, Gaussian mixture model, joint factor analysis, likelihood ratio, support vector machine, decision score, score fusion.

## 1 Introduction

Gaussian mixture model (GMM) has become one of the main building blocks of the speaker recognition technology. Here, each speaker is represented by the weights $w_i$, means $\mathbf{m}_i$, and covariances $\mathbf{\Sigma}_i$ of a mixture of $C$ multivariate Gaussian distribution defined in a $F$-dimensional continuous feature space. The well established practice in speaker recognition is not to train the parameters (usually only means) of the GMM independently for each speaker, but instead adapt them from the speaker-independent universal background model (UBM) [1].

The classical maximum a posteriori (MAP) algorithm [2], that is used for adapting the parameters of the target speaker model, seems to be the proper way when there is enough training data available and when there is no mismatch between training and testing (acoustic) conditions present. On the other hand, it does not try to address the problem of session variability, which remains one of the major challenges in text-independent speaker verification.

Various methods have been proposed to tackle the problem of session variability in the past [3,4,5,6,7]. Among them, the most general approach has been developed by Kenny et al. in a series of papers [8,9,10,11], where authors proposed a joint model of speaker and session variability in GMMs, dubbed joint factor analysis (JFA). Since then, JFA has become one of the corner stones in high-accuracy speaker recognition.

* The presented work has been partly financed by the European Union from the European Social Fund, contract No. 164/2009.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 139–146, 2011.

The paper is organized as follows. A theory behind JFA and SVM is briefly described in Section 2 and Section 3. In Section 4, a detailed description of the performed experiments is given. The evaluation results are presented in Section 5, followed by some brief conclusions in Section 6.

## 2 Joint Factor Analysis

The basic assumption in joint factor analysis is that speaker- and channel-dependent supervector $\mathbf{M}$ can be decomposed into a sum of two supervectors, a speaker supervector $\mathbf{s}$ and a channel supervector[1] $\mathbf{c}$:

$$\mathbf{M} = \mathbf{s} + \mathbf{c}, \tag{1}$$

where $\mathbf{s}$ and $\mathbf{c}$ are statistically independent and normally distributed.

To be able to carry out the decomposition, it turns out that we have to confine the channel-dependent supervector to lie in a low-dimensional subspace. This requirement seems reasonable, since the channel should not be able to transform one speaker into another, otherwise speaker recognition would be an ill-posed problem.

Kenny et al. proposed, that the first term in the right hand side of (1) should have a hidden variable description of the form

$$\mathbf{s} = \mathbf{m} + \mathbf{v}\mathbf{y} + \mathbf{d}\mathbf{z}, \tag{2}$$

where $\mathbf{m}$ is a speaker-independent supervector, $\mathbf{v}$ is a rectangular matrix of low rank, $\mathbf{d}$ is a diagonal matrix and $\mathbf{y}$ and $\mathbf{z}$ are normally distributed random vectors. The columns of $\mathbf{v}$ are referred as eigenvoices and the components of $\mathbf{y}$ as speaker factors.

Similarly, the second term of (1) has a hidden variable description of the form

$$\mathbf{c} = \mathbf{u}\mathbf{x}, \tag{3}$$

where $\mathbf{u}$ is a rectangular matrix of low rank and $\mathbf{x}$ is a normaly distributed random vector. The columns of $\mathbf{u}$ are reffered as eigenchannels and the components of $\mathbf{x}$ as channel factors.

The basic problem in joint factor analysis is the estimation of the hyper-parameters $\mathbf{m}$, $\mathbf{v}$, $\mathbf{d}$ and $\mathbf{u}$ on a large training set. For this task there have been maximum likelihood and maximum divergence estimation algorithms derived [8]. However, they are not suitable for implementation in the most general case, since they are both mathematicaly and computationaly demanding and their convergence is slow. Instead, there have been different variants of the algorithms proposed, where the hyperparameters get estimated in a decoupled fashion [11].

The underlying step in training the joint factor analysis model is the computation of the (joint) posterior distributions of the hidden variables $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. By writing $\mathbf{V} = (\mathbf{u}\,\mathbf{v}\,\mathbf{d})$, it has been shown that the posterior distribution of $\mathbf{X} = (\mathbf{x}\,\mathbf{y}\,\mathbf{z})^T$ is

[1] In JFA terminology, it is common to use the term supervector to refer to the $CF$-dimensional vector obtained by concatenating the $F$-dimensional mean vectors in the GMM, corresponding to a given utterance.

Gaussian with mean $\mathbf{L}^{-1}\mathbf{V}^T\boldsymbol{\Sigma}^{-1}(\mathbf{F}-\mathbf{Nm})$ and variance $\mathbf{L}^{-1}$, where $\mathbf{L}$ is a high-dimensional matrix:

$$\mathbf{L} = \mathbf{I} + \mathbf{V}^T\boldsymbol{\Sigma}^{-1}\mathbf{NV}. \tag{4}$$

Here, $\boldsymbol{\Sigma}$ is $CF \times CF$ covariance matrix whose diagonal is the concatenation of all the UBM's covariances; $\mathbf{N}$ is a $CF \times CF$ diagonal matrix whose diagonal blocks are $N_c\mathbf{I}$ with $N_c$ being the occupation count of $c$-th Gaussian; $\mathbf{F}$ is a $CF$-dimensional vector obtained by concatenating the first-order statistics for all Gaussians. These statistics are extracted from each utterance using the UBM.

The same posterior distribution needs to be computed for each speaker also at the enrollment time. To compute the score for each test utterance, the likelihood of the speaker's posterior distribution is compared with the prior distribution. In the fully Bayesian setting, the likelihood should be computed by marginalizing out all the hidden variables. Fortunately, it has been shown that using only the point estimates of the $\mathbf{y}$ and $\mathbf{z}$ suffices. (There is obviously enough training material available that the posterior distribution becomes sharply peaked.) This enable as to derive a fast likelihood ratio computation procedure, where only the integration over the channel distribution is performed [9].

This can be written as

$$P(\mathcal{X}|\mathbf{s}) = \int P(\mathcal{X}|\mathbf{s},\mathbf{x})\,\mathcal{N}(\mathbf{x}|0,I)\,\mathrm{d}\mathbf{x}, \tag{5}$$

where $\mathcal{X}$ stands for the collection of the feature vectors from the test utterance.

If a fixed alignment of the feature vectors to the Gaussian components is assumed, (lower bound of) the integral in (5) can be solved in a closed form:

$$\begin{aligned}\log P(\mathcal{X}|\mathbf{s}) = &\sum_{c=1}^{C} N_c \log \frac{w_c}{(2\pi)^{F/2}|\boldsymbol{\Sigma}_c|^{1/2}}\\ &-\frac{1}{2}\mathrm{t}r(\boldsymbol{\Sigma}^{-1}\mathbf{S}_s) - \frac{1}{2}\log|\mathbf{l}|\\ &+\frac{1}{2}||\mathbf{l}^{-1/2}\mathbf{u}^T\boldsymbol{\Sigma}^{-1}\mathbf{F}_s||^2,\end{aligned} \tag{6}$$

where $\mathbf{F}_s$ and $\mathbf{S}_s$ are the centralized first- and the second-order Baum-Welch statistics and l is a matrix given by $\mathbf{l} = \mathbf{I} + \mathbf{u}^T\boldsymbol{\Sigma}^{-1}\mathbf{Nu}$.

## 3 Support Vector Machine

Support vector machine (SVM) is a two-class classifier, based on the concept of the maximum margin. Given two sets of data points, the SVM finds a separating hyperplane which has the maximum distance to the nearest data point (i.e., maximum margin). It turns out that the separating hyperplane is only a function of the training data that lie on the margin (these are called the support vectors).

The classification of the test vector is as simple as determine on which side of the hyperplane (decision boundary) the given test vector lies. The SVM decision function can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b, \tag{7}$$

with the constraints $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $\alpha_i > 0$. The $y_i$ are target values (either -1 or 1, depending on which class the corresponding support vector $\mathbf{x}_i$ comes from).

Although the SVM, as originally proposed, is a linear classifier, it has been later generalized to the non-linear case. This was achieved through the introduction of the so called kernel trick, which enable us to do classification in a high (possibly infinite) dimensional feature space – where the data becomes linearly separable – without ever explicitly mapping the data to that space. This is possible since the data in the training problem appears only in the form of dot products. So by replacing the dot product $\mathbf{x}_i\mathbf{x}_j$ with a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$, we implicitly map the data to some (kernel-induced) feature space.

One example is Gaussian kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{x}_j||^2\right), \tag{8}$$

for which the corresponding feature space is a Hilbert space of infinite dimension.

The SVMs have been extensively used in speaker verification community [5,6,7,12,13,14]. However, they are usually applied in combination with the nuisance attribute projection method (NAP) of channel compensation [5], and not with the JFA, as in our case.

## 4 Experimental Setup

### 4.1 Test Set

The results of our experiments are reported on the telephone part of the core condition of the NIST 2008 speaker recognition evaluation (SRE) dataset [15].

### 4.2 Feature Extraction

In our experiments, we used cepstral features, extracted from a speech signal using a 25 ms Hamming window. 19 mel frequency cepstral coefficients together with log energy were calculated every 10 ms. This 20-dimensional feature vectors were subjected to feature warping [16] using a 3 s sliding window. Delta and double delta coefficients were then calculated using a 5 frames window yielding 60-dimensional feature vectors.

To suppress silences, we did not use our own silence detector, but instead relied on the time stamps provided by NIST.

### 4.3 Factor Analysis Training

We used two gender-dependent universal background models (UBMs), each containing 512 Gaussian components. These UBMs were trained using the telephone speech data from the NIST 2004-2006 SRE collection.

The two gender dependent factor analysis models were trained on the same data as the UBMs. To avoid the overfitting problem of the ML estimation, we estimated $\mathbf{v}$ and $\mathbf{d}$ in a decoupled fashion proposed as by Kenny et al. [11]. The eigenvoice matrix $\mathbf{v}$ was estimated on the NIST 2005 SRE and NIST 2006 SRE data, while the diagonal matrix $\mathbf{d}$ was estimated on the NIST 2004 SRE data. The number of eigenvoices was set to 300. To be able to ignore the channel effects while estimating $\mathbf{v}$ and $\mathbf{d}$, we considered only those speakers for which five or more recordings were available. (It is reasonable to believe that by pooling together sufficiently many recordings of the same speaker, the channel effects get averaged out.)

### 4.4 Estimation of the Speaker Model

At the enrollment time we estimated the posterior distribution of the variables $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, as described in Section 2, but only the mean of the speaker supervector $\mathbf{s}$,

$$\mathbf{s} = \mathbf{m} + \mathbf{v}\mathbb{E}[\mathbf{y}] + \mathbf{d}\mathbb{E}[\mathbf{z}], \tag{9}$$

was kept. The speaker model was therefore presented as a 30720-dimensional supervector.

### 4.5 Scoring

We implemented two types of scoring methods. One based on the likelihood ratio statistics, the other on support vector machines decision values.

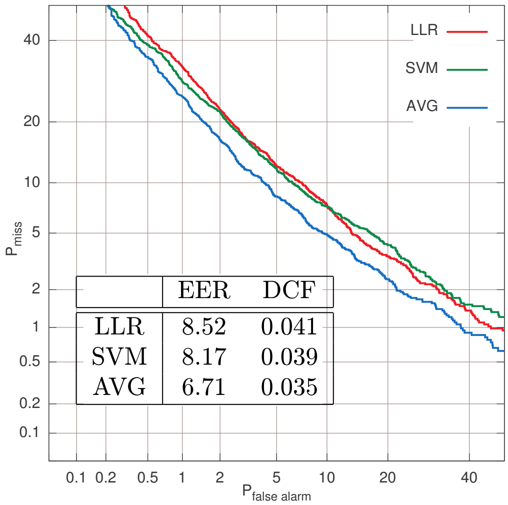
**Likelihood Ratio.** For the given test utterance $\mathcal{X}$ and for the given hypothesized speaker $\mathbf{s}$, the LR[2] is computed as the ratio between two likelihoods:

$$\frac{P(\mathcal{X}|\mathbf{s})}{P(\mathcal{X}|\mathbf{m})}, \tag{10}$$

where $\mathbf{s}$ and $\mathbf{m}$ are speaker-dependent and speaker-independent supervectors, respectively, and $\mathcal{X}$ is the collection of short-time feature vectors, extracted from the test utterance. The two likelihoods are computed using the expression (6).

**Support Vector Machine.** In contrast with LR, where only the training utterance is converted to a supervector, here both training and test utterance have to be mapped to a supervector space prior to calculating the score. The resulting supervectors should be appropriately preprocessed. Usually, a simple linear scaling is adopted for this task. However, we decided to use the rank normalization algorithm [17], where each feature value is replaced by its rank in the background data, followed by a normalization to the unit interval. The background data in our case consisted of 1000 randomly selected utterances from the NIST 2005 SRE database.

[2] In practice, we take the logarithm of LR, dividing it by the length of the test utterance, to compensate for different lengths of the recordings.



| | EER | DCF |
|---|---|---|
| LLR | 8.52 | 0.041 |
| SVM | 8.17 | 0.039 |
| AVG | 6.71 | 0.035 |

**Fig. 1.** DET plots for log-likelihood ratio (LLR), support vector machine (SVM) and the average of both (AVG), obtained on the NIST 2008 SRE (core condition, telephone part)

After the preprocessing, we need first to find the separating hyperplane between the training supervector and the background supervectors. Then the SVM-based decision score for the test utterance can be computed using (7).

We used Gaussian kernel (8) in our experiments. The kernel parameter $\sigma$ was set to 0.001. For the background data we reused the same utterances that have been involved in the rank normalization.

### 4.6 Score Normalization

We tried different score normalizations [18], namely z-norm, t-norm, zt-norm and tz-norm. We found out, that for LR-based scores, the zt-norm is the most effective method, as expected. On the other hand, score normalization was not beneficial for the SVM-based scores. This came as a little surprise to us, since most of the published results uses t-norm normalization for SVM-based scores.

### 4.7 Score Fusion

Score fusion, when applied to scores produced by different systems, usually boosts the performance of the individual systems. However, this was not self-evident in our case, since our scores were produced by two very similar systems, which differed only in the decision criterion used for score calculation.

In the presented work we combined scrores by a simple linear fusion, although also other more sophisticated fusion methods (e.g. logistic regression, which was used in [19]) could be used.

## 5 Results

Fig. 1 shows the results for the two types of scoring methods (LLR and SVM). Although the SVM-based scoring gives slightly better results, the difference is not significant. However, if we compute the average of both scores (AVG), the performance is substantially improved. Comparing the performance of the AVG system to the SVM system, we see, that the EER drops from 8.17% to 6.71% (22% relative improvement) and the DCF drops from 0.039 to 0.035 (11% relative improvement).

## 6 Conclusions

It is well known that fusion of heterogenous speaker verification systems can substantially improve the performance of individual systems [19]. However, a lot of time and effort is needed to implement many different systems. On the other hand, it is relatively easy to implement different scoring methods on top of the same system, but the scores may be too correlated for the fusion to be effective.

Interestingly, we have shown that generative (LR) and discriminative (SVM) scoring methods contribute enough complementary information, since we achieved a considerable improvement in verification performance by a simple linear fusion of both scores.

Recently, there has been a shift from supervector-based systems (an example of such systems is described in this paper) towards i-vector based systems observed [20]. Therefore it would be interesting to see whether a similar complementarity between different decision criteria can be observed also in the i-vector space. However, that question remains to be answered in the future.

## References

1. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted Gaussian mixture models. Digital Signal Processing 10(1), 19–41 (2000)
2. Gauvain, J.-L., Lee, C.-H.: Maximum a posteriori estimation for multivariate Gaussian mixtureobservations of Markov chains. IEEE Trans. Speech and Audio Processing 2(2), 291–298 (1994)
3. Reynolds, D.A.: Channel robust speaker verification via feature mapping. In: Proc. ICASSP 2003, Hong Kong, vol. 2, pp. 53–56 (April 2003)
4. Teunen, R., Shahshahani, B., Heck, L.: A model-based transformational approach to robust speaker recognition. In: Proc. ICSLP 2000, Beijing, China, pp. 495–498 (October 2000)
5. Solomonoff, A., Quillen, C., Campbell, W.M.: Channel compensation for SVM speaker recognition. In: A Speaker Odyssey: The Speaker Recognition Workshop, Toledo, Spain, pp. 41–44 (May-June 2004)
6. Stolcke, A., Kajarekar, S.S., Ferrer, L., Shriberg, E.: Speaker recognition with session variability normalization based on MLLR adaptation transforms. IEEE Trans. Audio, Speech and Lang. Proc. 15(7), 1987–1998 (2007)
7. Hatch, A.O., Kajarekar, S., Stolcke, A.: Within-class covariance normalization for SVM-based speaker recognition. In: Proc. INTERSPEECH/ICSLP 2006, Pittsburgh, USA, vol. 3, pp. 1471–1474 (September 2006)
8. Kenny, P.: Joint factor analysis of speaker and session variability: Theory and algorithms, technical report CRIM-06/08-13. CRIM, Montreal (2005)

9. Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P.: Joint factor analysis versus eigenchannels in speaker recognition. IEEE Trans. Audio Speech and Lang. Process. 15(4), 1435–1447 (2007)
10. Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P.: Speaker and session variability in GMM-based speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 15(4), 1448–1460 (2007)
11. Kenny, P., Ouellet, P., Dehak, N., Gupta, V., Dumouchel, P.: A study of inter-speaker variability in speaker verification. IEEE Trans. Audio, Speech and Language Processing 16(5), 980–988 (2008)
12. Campbell, W.M., Sturim, D.E., Reynolds, D.A.: Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters 13(5), 308–311 (2006)
13. Campbell, W.M., Sturim, D.E., Reynolds, D.A., Solomonoff, A.: SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In: Proc. ICASSP 2006, Toulouse, France, vol. 1, pp. 97–100 (May 2006)
14. Campbell, W.M.: Generalized linear discriminant sequence kernels for speaker recognition. In: Proc. ICASSP 2002, Orlando, vol. 1, pp. 161–164 (2002)
15. NIST Speaker Recognition Evaluation (2008), http://www.itl.nist.gov/iad/mig/tests/sre/2008/index.html
16. Pelecanos, J., Sridharan, S.: Feature warping for robust speaker verification. In: Odyssey 2001, Crete, Greece, pp. 213–218 (June 2001)
17. Stolcke, A., Kajarekar, S., Ferrer, L.: Nonparametric Feature Normalization for SVM-Based Speaker Verification. In: Proc. ICASSP 2008, Las Vegas, Nevada, pp. 1577–1580 (2008)
18. Auckenthaler, R., Carey, M., Lloyd-Thomas, H.: Score normalization for text-independent speaker verification systems. Digital Signal Processing 10(1), 42–54 (2000)
19. Brummer, N., Burget, L., Černocky, J., Glembek, O., Grezl, F., Karafiat, M., van Leeuwen, D., Matejka, P., Schwarz, P., Strasheim, A.: Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006. IEEE Trans. Audio, Speech, and Language Processing 15(7), 2072–2084 (2007)
20. Kenny, P.: Bayesian Speaker Verification with Heavy-Tailed Priors. In: Proc. Odyssey, Brno, Czech Republic (July 2010)

# Generalized Non-uniform Time Scaling Distribution Method for Natural-Sounding Speech Rate Change*

Daniel Tihelka and Martin Méner

University of West Bohemia, Faculty of Applied Sciences, Department of Cybernetics,
Univerzitní 8, 306 14 Plzeň, Czech Republic
dtihelka@kky.zcu.cz, mmener@students.zcu.cz

**Abstract.** The paper proposes a general, flexible and efficient method for the distribution of time-scale modification Factors. The method is inspired by the analogy with a sequence of springs with different rates/constants, allowing simple and straightforward non-linear distribution of modification factors through the speech to modify. The flexibility and generality of the proposed scheme enables its use for any number of speech/sound segment categories of any type and length, while the modification factors can either be set heuristically, ad-hoc, or trained from data. At the end of the paper, an attempt to use statistics of phone durations to set the modification factors is described and discussed.

**Keywords:** non-uniform time scaling, WSOLA, speech synthesis, spring sequence, spring rate.

## 1 Introduction

It is desirable to have a method capable of transforming natural-sounding speech into natural-sounding speech in higher/lower tempo. For example, unit selection, which simply concatenates raw unit waveforms without any further signal processing, produces close-to-natural sounding speech, but with fixed timing (in the sense of “fixed to the style of source speaker”). It is limiting for many potential applications – for example, visually impaired people usually prefer much faster-than-natural sounding speech (to speed up the transfer of information; they are trained to understand speech otherwise almost unintelligible), while they also want the speech to sound natural. In contrast, it may be required to slow down the speech generated from a corpus recorded by a “fast jabbering” speaker. Of course, there are more methods capable of doing this, e.g. [1,2,3], but the most widely adopted is Waveform Similarity OverLap-Add – WSOLA [4].

The technique is usually implemented with uniform time-scaling factors, i.e. each part of the entire signal is stretched or compressed by the same factor. However, it has been pointed-out in many studies, e.g. [5,6,7,8], that a uniform time-scaled speech signal can sound unnatural and difficult to comprehend: with time-stretching, phones

[*] This research was supported 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 by the internal UWB grant, project No. SGS-2010-054.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 147–154, 2011.

and intonation can become dull and with time-compression, important speech characteristics can disappear. Therefore, this paper extends and generalizes the proposals of the non-linear scale factor distribution introduced in the above-mentioned papers, making them as flexible as possible. The new scheme of assigning particular stretch/compression factor to individual speech segments is based on the analogy with a sequence of springs with different rate/constants.

## 2 Time-Scale Modifications of Speech

WSOLA is based on Overlap-add technique [9] with the selection of overlapping segments aiming to maintain sufficient signal continuity at overlapping segment joins by maximizing similarity to the natural continuity that existed in the input original waveform. It attempts to keep pitch and phase relationships in the original waveform, as phase jumps and pitch-period distortions would otherwise cause audible unnatural artefacts, as shown in Figure 1.

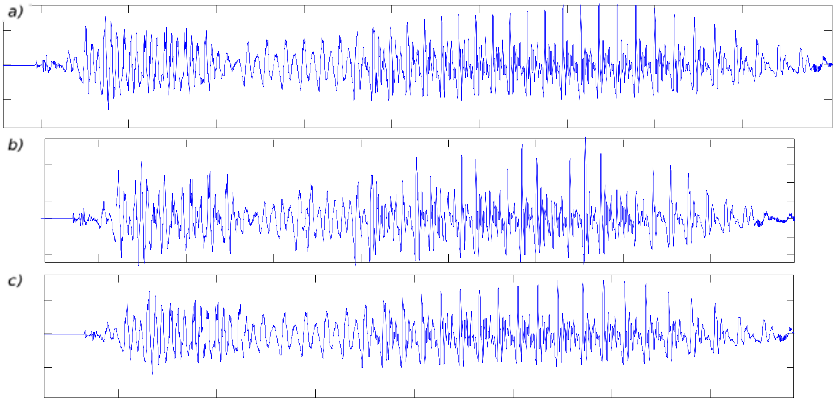


**Fig. 1.** The illustration of phase and pitch distortions caused by OLA method; *a)* is the original waveform, *b)* is the OLA-modified and *c)* is the WSOLA-modified version of it. Both were $1.15\times$ compressed, using approximately 3 pitch-period window.

Most commonly, the time stretching is carried out in a uniform way, meaning that each part of the whole signal is modified by the same factor. However, as far as speech is concerned, for higher stretch of compression ratios even WSOLA may cause certain local speech quality degradations – those are mainly the doubling (or loss, respectively) of significant short-term speech characteristics, like the release/burst parts of plosives, glottal stops, or phone transitions. According to our experience, such phenomena start to appear when speech is uniform time-scaled by more than 20–25%, as illustrated by Figure 2. It is clear (as has been demonstrated many times) that there are segments in speech which can be modified more that others without effectively changing the intelligibility or naturalness. However, such non-uniform scaling requires some knowledge of individual phones and their boundaries in the original speech waveform.

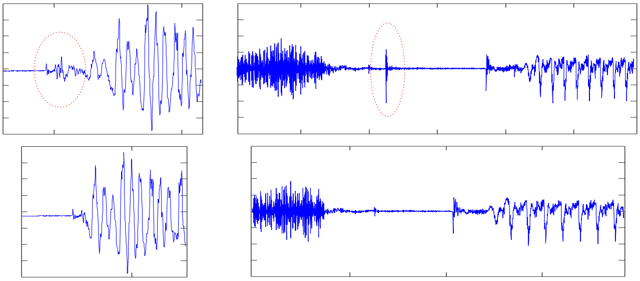


**Fig. 2.** The illustration of plosive's release part deletions caused by WSOLA when uniformly modified by factor 0.8. It affects both intelligibility (phone is missing) and naturalness (the second plosive sounds rather long).

The authors in [6] avoided the need of exact phones identity knowledge by estimating *audio tension*, computed as the combination of local emphasis and speaking rate given by a rather complex signal analysis. The order of allowed signal modification is then in inverse relation to the tension. The authors in [5] categorised speech into transit/steady segments comparing LPC cepstral distances. Similarly, [7,8] also use signal analysis, albeit different, to categorise the speech into 5 independent categories, with individual modification factor defined for each category. What the studies share is that they all work with raw speech stream without exact phone boundaries marked. Therefore, the authors used fixed modification factors for individual segments. The main problem with this, however, is that such an approach cannot ensure the required overall modification factor. In [6], slow-response feedback loop adjusting the actual modification rate was used. Of course, it significantly complicates the scaling process itself as well as its tuning (too slow does not guarantee reaching the overall factor, not slow enough may lead to rapid changes manifested by unnatural artefacts).

The situation is much easier in the case of scaling TTS-generated speech, since there is precise marking of phone boundaries (and thus transits as well) available. Moreover, as TTS usually generates speech in chunks, it is possible to optimize distribution of non-linear modification factors through the chunk. Able to work witch such chunks, the proposed technique can be used to optimize the distribution of modification factors with the guarantee of reaching the required overall rate. The proposed scheme is also very general, enabling work with any number of segments of any length and type – either they are individual phones or any kind of sound categories detected in the studies mentioned. Due to the flexibility of the proposed algorithm, the categorisation may be narrowed or widened dynamically, without the need of any special adjustments (which is not the case of the other studies).

## 3 Generalized Non-linear Scaling Scheme

To keep the description universal, let *segment* denote a part of signal which can be stretched or compressed linearly (e.g. a phone). The idea is based on the analogy with a spring — each speech segment can be viewed as a spring with a stiffness (or stretchability) characterised by the *rate/constant* value[1]. The rate is an analogy to the units tolerance of stretching or compression – the lower the rate is set, the more the segment can be modified without the loss of its important characteristics. The sequence of segments to modify can then be viewed as a sequence of springs of different lengths and different rates. For the required modification factor, analogous to force attempting to stretch or compress the sequence of springs, each segments will be modified by a different ratio corresponding to its rate — see Figure 3.

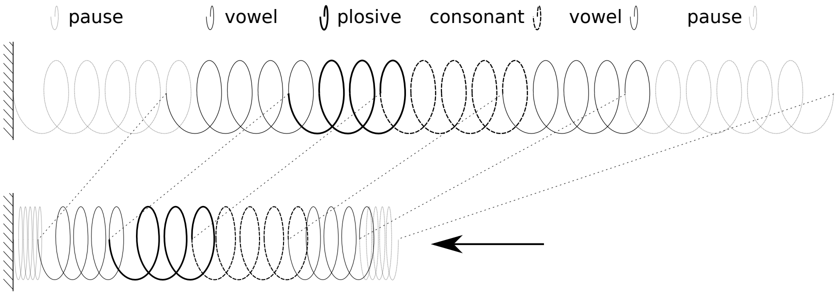


**Fig. 3.** The illustration of non-linear scaling of phones with various tolerance to stretching/compression, viewed as sequence of springs with different rates/constants

To define it formally, let $r_i$ be the rate coefficient set for $i$-th segment in the sequence of segments to modify, and $x_i$ be the original length of the segment. Having modification factor $f_i$ (unknown yet), the new segment length will be $x'_i = f_i x_i$. Regarding the spring analogy, $d_i = x_i - x'_i$ represents the displacement of the spring's end from its equilibrium position. Extending it for the sequence of $i = 1, 2, \ldots, I$ segments/springs, the rate of the whole sequence $r$ is

$$\frac{1}{r} = \sum_{i}^{I} \frac{1}{r_i} \tag{1}$$

According to Hooke's law of elasticity[2], the restoring power $F$ of the sequence is given as

$$F = -rd \tag{2}$$

where for $f$ being the required overall modification factor is

$$d = fx - x = x' - x = \sum_{i}^{I} (x'_i - x_i) \tag{3}$$

[1] We will use the term “rate” further in text.

[2] The approximation stating that the extension of a spring is in direct proportion with the force affecting it.

Having the power of the whole sequence, we can simply compute the deformation $d_i$ for each $i$-th segment as

$$d_i = \frac{F}{r_i} \qquad i = 1, \ldots, I \tag{4}$$

and use standard WSOLA to modify the segment by the factor $f_i = x'_i / x_i$.

In reality (and it is clear from the equations), the deformation of a spring does not depend on its length. It means that both $i$th segment with duration $x_i$, and $j$th segment with duration $x_j = n x_i$ will be deformed by $d = d_i = d_j$, when $r_i = r_j$, and thus $f_i \neq f_j$. To avoid this, i.e. to reach $f_i = f_j$, which is naturally expected in case of speech modification, we replaced Equation 1 by

$$\frac{1}{r} = \sum_{i}^{I} \frac{1}{\frac{x r_i}{x_i}} \tag{5}$$

effectively normalizing the pre-set $r_i$ by the duration of segment (the shorter the segment is, the higher rate it has).

This scheme will work fairly well for smaller overall stretching factors $f$. However, under a higher pressure, the Equation 4 may lead to $d_i > x_i$, meaning that the spring's right end is moved before its left end, if pressed from right. The solution is to reflect the increasing resistance of springs when deformed, and balance the overall deformation until force equilibrium is reached. For our purpose, we, nevertheless, composed a much simpler iteration scheme, as the efficient approximation. We define the minimum length of segments (or segment categories) $x^*_i$, and check the computed modification against this minimum value:

1. initialize $\mathcal{I} = \{1, \ldots, I\}$
2. compute $x'_i \quad \forall i \in \mathcal{I}$
3. find $x'_i \leq x^*_i$, store such $i$ in $\overline{\mathcal{I}}$
4. if $\overline{\mathcal{I}} = \{\}$, compute $f_i \quad \forall i = 1, \ldots, I$ and carry out segments modification. Otherwise continue to step 5.
5. fix $x'_i = x^*_i \quad \forall i \in \overline{\mathcal{I}}$
6. update $\mathcal{I} = \mathcal{I} - \overline{\mathcal{I}}$
7. if $\mathcal{I} = \{\}$, compute $f_i \quad \forall i = 1, \ldots, I$ and carry out segments modification (all segments were modified to $x^*_i$).
8. update $d = \sum\limits_{i \in \mathcal{I}} x_i + \sum\limits_{i \in \overline{\mathcal{I}}} x'_i$ and continue by step 2.

Similarly, the algorithm can simply be modified not to allow the exceeding of a maximum length $x^\star_i$. The advantage of such algorithm is that it iterates only until there are "over-modified" segments. In the worst case, the number of iterations equals to the number of different rates.

The proposed scheme is extremely flexible, since it does not place any requirements or constraints on what actually segment should be or how long it should be. The segment may be a phone, a category of phones, or even phone's transition region (as also used in [8]). Moreover, no matter how many segment types are defined, the algorithm will still remain the same. Also, the rate factors may either be set heuristically, or tuned by hand, or obtained from data in speaker-dependent manner.

### 3.1 Scheme Illustration

Let us consider, for the following illustration, the categorising of phones with their rates $r$ set to:

| | | |
|---|---|---|
| *pauses* | | 0.1 |
| *short vowels* | [i,e,a,o,u] | 0.4 |
| *long vowels and diphthongs* | [i:,e:,a:,o:,u:,0_u,a_u,e_u] | 0.5 |
| *fricatives and affricates* | [d_z,d_Z,t_s,t_S,P\,Q\, x,f,s,S,z,Z,h\,G] | 0.6 |
| *nasals, glides and liquids* | [m,n,N,J,M,j,l,L,H,v] | 0.6 |
| *'r', plosives and glottal stop* | [r,p,b,t,d,c,J\,k,g,?] | 1.0 |

Note that 1 does not mean *do not modify this category*, the value only means that the category is about 2.5× more stiff than vowels, for example. If a category should not be modified, its rate can simply be set to a relatively high value.

Let us further have an artificial sequence [*pause vowel plosive nasal vowel pause*] to modify, with the minimum length $x^* = 0.01$. Table 1 illustrates the non-uniform stretching of the whole sequence for the given original segment lengths and various modification factors.

**Table 1.** The illustration of non-uniform time stretching for the sequence of segments with various rates; *orig.* column contains the original (artificial) length of the particular segment

| | orig. | *factor/new length* for overall factor | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.3 | 0.5 | 0.8 | 1.25 | 2.0 | 3.3 |
| *pause* | 33.0 | 0.00/ 0.01 | 0.19/ 6.4 | 0.68/ 22.3 | 1.40/46.3 | 2.62/ 86.3 | 4.77/157.3 |
| *vowel* | 25.0 | 0.55/13.64 | 0.80/20.0 | 0.92/ 23.0 | 1.10/27.5 | 1.40/ 35.1 | 1.94/ 48.6 |
| *plosive* | 15.0 | 0.82/12.27 | 0.92/13.8 | 0.97/ 14.5 | 1.04/15.6 | 1.16/ 17.4 | 1.38/ 20.7 |
| *nasal* | 10.0 | 0.70/ 6.97 | 0.87/ 8.7 | 0.95/ 9.5 | 1.07/10.7 | 1.27/ 12.7 | 1.63/ 16.3 |
| *vowel* | 20.0 | 0.55/10.91 | 0.80/16.0 | 0.92/ 18.4 | 1.10/22.0 | 1.40/ 28.1 | 1.94/ 38.8 |
| *pause* | 43.0 | 0.00/ 0.01 | 0.19/ 8.3 | 0.68/ 29.1 | 1.40/60.4 | 2.62/112.4 | 4.77/205.0 |

In this example, we do not consider phone transitions which belong to one of categories in [5,8]. However, it can easily be added by considering a region around phone boundaries with high rate value.

## 4 Setting the Rates

Having the general method distributing modifications through individual segments, the key point is now to set the rate values for individual segment categories. Of course, the inspiration may be taken from [6] or [8]. However, such approaches add significant

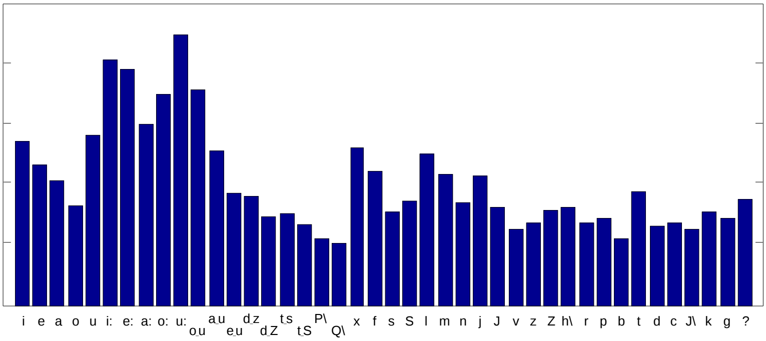


**Fig. 4.** The illustration of phone duration variances for one of Czech voices; phones are in SAMPA alphabet. The remaining Czech voices displayed roughly similar character.

amount of signal processing in the case where we have exact phone boundaries at our disposal. Our initial idea, therefore, was to base the rates on the natural variability of speech — the more the length of a phone differs in natural speech, the lower rate value for the phone can be set (it is less stiff naturally). In addition, taking rates from speech analysis would enable fine-tuning the rates for individual voices without any effort, which is virtually impossible for signal analysis-based techniques.

The variance of phone durations was computed from the automatic segmentation [11] of 4 Czech corpora with the belief that the relations of the variances will be directly transformable into the rates for the individual phones or some categories. All the corpora were quite large, containing at least 15 hours of speech (excluding pauses). Unfortunately, it is illustrated on Figure 4 that such a statistic is not a good model for the non-linear scaling configuration — for example, the plosives have roughly the same duration variance as short vowels, which would lead to similar modification rates (causing problems discussed in Section 2). The only exception are long vowels, possessing measurably higher variance, but allowing a lower rate for them will basically lead to their unification with short vowels, when compressed. The reason of such observation is most likely in computing the statistics from corpus with fairly uniform speech tempo. Instead, we would need to compare the characteristics of a speaker speaking with different (both lower and higher, ideally) tempo, which we do not have yet. For now, we therefore have had to set the rates heuristically to values shown in example in Section 3.1.

## 5 Conclusion

The proposed non-linear time scale distribution scheme allows us to assign the rate (level of non-readiness to modify) to any segment of speech (or sound, in general), and computes individual stretching/compression factors of the particular segments needed to reach the required overall modification factor. There are no constraints for segment type, length, number or their distribution through speech chunk to modify. In our work on [10], we considered the segments equal to phone categories with hand-tuned rate values, and modified them using WSOLA technique (though any other time-scale modification technique can be used).

What remains to be completed is a more formal evaluation of modified speech quality. However, our listening during testing as well as the evidence in the other papers convinces us that there will not be any hidden surprises.

## References

1. Huang, Y., Xu, B.: A novel model TD-PSPTP for speech synthesis. In: Proc. EUROSPEECH 1999, Budapest, Hungary, pp. 2303–2306 (1999)
2. Lawlor, B., Fagan, A.D.: A novel high quality efficient algorithm for time-scale modification of speech. In: Proc. EUROSPEECH 1999, Budapest, Hungary, pp. 2785–2788 (1999)
3. Kumar, S., Mandal, D., Datta, A.K.: Epoch Synchronous Non Overlap Add (ESNOLA) Method based Concatenative Speech Synthesis System for Bangla. In: Proc. SSW–6, Bonn, Germany (2007)
4. Verhelst, W., Roelands, M.: An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High Qquality Time-Scale Modification of Speech. In: Proc. ICASSP, Minneapolis, USA, vol. 2, pp. 554–557 (1993)
5. Lee, S., Kim, H.D., Kim, H.S.: Variable Time-Scale Modification of Speech Using Transient Information. In: Proc. ICASSP, Munich, Germany, vol. 1, pp. 1319–1322 (1997)
6. Covell, M., Withgott, M., Slaney, M.: MACH1: Non Uniform Time-Scale Modification of Speech. In: Proc. ICASSP, Seattle, USA, vol. 1, pp. 349–352 (1998)
7. Donnellan, O., Jung, E., Coylem, E.: Speech-Adaptive Time-scale Modification for Computer Assited Language Learning. In: Proc. ICALT 2003, Aix-en-Provence, France, pp. 165–169 (2003)
8. Demol, M., Verhelst, W., Struyve, K., Verhoeve, P.: Efcient non-uniform time-scaling of speech with WSOLA. In: Proc. SPECOM, Patras, Greece, pp. 163–166 (2005)
9. Verhelst, W.: Overlap-add Methods for Time-Scaling of Speech. Speech Communication 30(4), 207–221 (2000)
10. Hanzlíček, Z., Matoušek, J., Tihelka, D.: Towards automatic audio track generation for Czech TV broadcasting: Initial experiments with subtitles-to-speech synthesis. In: Proc. ICSP, Beijing, China, vol. 3, pp. 2721–2724 (2008)
11. Matoušek, J., Romportl, J.: Automatic Pitch-Synchronous Phonetic Segmentation. In: Proc. Interspeech, Brisbane, Australia, pp. 1626–1629 (2008)

# Grouping Alternating Schemata in Semantic Valence Dictionary of Polish Verbs

Elżbieta Hajnicz

Institute of Computer Science, Polish Academy of Sciences

**Abstract.** In this paper a method for grouping semantically related schemata is proposed. It is based on the participation of verbs in diathesis alternations. The process of aggregating schemata in four steps is presented.

## 1 Introduction

The primary task of our research is to create a semantic valence dictionary in an automatic way. To accomplish this goal, the syntactic valence dictionary of Polish verbs is supplemented with semantic information, provided by wordnet's semantic categories [5] or synsets [6] of nouns. In our present work we focus on slots being nominal phrases NPs and prepositional-nominal phrases PrepNPs, whose semantic heads are nouns. We discuss the case of 25 predefined semantic categories of nouns.

In our previous works [7], [8] we focused on the preparation of a syntactic-semantic valence dictionary in which each slot of a syntactic schema[1] is supplied with a list of semantic categories, forming a semantic frame. However, a genuine semantic dictionary is composed of semantic frames represented as predicate-argument structure, in spite of its syntactic realisations. Each semantic argument is connected with its semantic role. On the other hand, syntactic slots of schemata are provided with information about which semantic role they realise.

We say that semantically related schemata participate in diathesis alternation. In [9] a method of detecting diathesis alternation on the basis of semantic similarity between schemata is presented. In the current paper we propose a method of using this information in order to group related schemata. This is performed concurrently with linking their arguments, which is described in [10].

## 2 Related Works

There exist several manually prepared semantic valence dictionaries. For English, the most famous are FrameNet, VerbNet and PropBank. FrameNet [2] is based on Fillmore's frame semantics. It contains a hierarchy of frames consisting of sets of semantic roles. Particular predicates evoke corresponding frames, and slots of their syntactic schemata are linked to corresponding roles. VerbNet [14] implements Levin's [17] classification of verbs based on their alternation behaviour. Valence information is represented as *Lexicalized Tree Adjoining Grammar* trees having nodes labelled with general

[1] We use the term syntactic *schema* instead of very popular syntactic *frame* in order to distinguish it from the term *semantic frame*.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 155–162, 2011.

syntactic roles and Princeton WordNet based selectional restrictions. Special operations on trees serve to link trees representing different schemata participating in a corresponding alternation. The goal of PropBank [23] was to extend Penn Tree Bank [19] with semantic valence information and validate Levin's classification using it. Each verb (in a particular sense) has its purely semantic frame in a form of a list of arguments numbered from Arg0 up to (potentially) Arg5 and labelled with semantic roles, generally verb-specific. Adjuncts are denoted ArgM and have general labels, such as TMP for time or LOC for location.

In the case of Slavic languages, there exist two extensive dictionaries for Czech, both having specific frames for each verb. VALLEX [28] is connected with Prague Dependency Treebank [4]. A valence frame consists of (obligatory or optional) inner participants and typical free modifications. Each slot is equipped with a functor (a deep role), a list of possible morphosyntactic realisations and a complementation type (obligatory, optional—for inner participants or typical—for free modifications). Additional information concerns reflexivity, reciprocity and control. Verb senses are grouped into semantic classes. VerbaLex [12] was designed as a valence extension of Czech WordNet [22]. Valence frames are assigned to wordnet classes, not to individual verbs. Frames are quite similar to these of VALLEX. However, the set of roles is different and it has two level of granularity. They are based on EuroWordNet Top Ontology and constitute selectional preferences rather than typical semantic roles.

All these dictionaries were prepared manually. Automatic processing related to valence focuses on selectional preferences [18], [24], [1], alternation detection [15], [20], or verb classification [3], [16], [25], [21], [13]. Grouping semantically related schemata of a verb can be a side-effect of the suggested procedures, but to our best knowledge no work has been dedicated to the subject of grouping schemata.

## 3 Valence Dictionary

A syntactic valence dictionary $\mathcal{D}$ is a set of entries representing schemata for every verb considered. Formally, $\mathcal{D}$ is a set of pairs $\langle v, g\rangle$, where $v \in V$ is a verb and a syntactic schema $g \in G$ is a set of slots. The set of syntactic schemata of a particular verb $v$ will be denoted as $G_v$. The dictionary of 32 verbs chosen for the experiment was prepared on the basis of Świdziński's [27] dictionary. Verbs were chosen manually in a way to maximise the variability of their syntactic frames (in particular, diathesis alternations) on one hand and the polysemy within a single frame on the other. Their frequency was an important criterion for this choice as well.

The list of slots can include: adjectival phrases (AdjP), adverbial phrases (AdvP), infinitival phrases (InfP), nominal phrases (NP), prepositional-adjectival phrases (PrepAdjP), prepositional-nominal phrases (PrepNP) and clauses (SentP). A special slot sie hosts the reflexive marker. Some slots are parametrised. In particular, the only parameter o NP is its case, whereas PrepNP has two parameters: the form of a preposition and the case of its NP complement.

## 4 Classification of Alternations

For Polish, there is no comprehensive classification of diathesis alternations and verbs participating in them, as [17] serves for English. Szupryczyńska [26] conducted the analysis of accusative verbs.

In [9] we presented a very coarse purely syntactic classification of potential alternations, describing only how the alternation relates slots in two schemata $g^A$ and $g^B$ involved in it. The alternations can be divided into two types. First, there are alternations preserving the number of arguments in both schemata. This condition is satisfied by alternations referred to in [9] as *simple* alternation, exemplified by dative alternation, see (1), *cross* alternation exemplified by locative alternation, see (2), *simple reflexive* alternation, cf. (5), and *cross-reflexive*, cf. (6). Second, there are alternations in which one of the alternating slots is absent in one schema. This condition is satisfied by alternations referred to in [9] as *deletion* alternation exemplified by object drop alternation, see (3), *shift* alternation exemplified by an unreflexive case of causative alternation, see (4). *Reflexive deletion* alternation is exemplified by reflexive alternation, where the reflexive marker *się* plays the role of *oneself*, cf. (7), and reciprocal alternation, where reflexive marker *się* plays the role of *each other*, cf. (8).[2] *Reflexive shift* alternation is exemplified by causative alternation, cf. (9).

(1) *Chłopak posłał książkę koledze. (A boy sent his friend a book.)*
*Chłopak posłał książkę do kolegi. (A boy sent a book to his friend.)*
(2) *Rolnik załadował wóz sianem. (The farmer loaded the wagon with hay.)*
*Rolnik załadował siano na wóz. (The farmer loaded hay onto the wagon.)*
(3) *Matka pozmywała naczynia. (Mother washed dishes.)*
*Matka pozmywała. (Mother washed.)*
(4) *Jeździec pognał konia przez las. (The rider rode a horse across a forest.)*
*Koń pognał przez las. (A horse rode across a forest.)*
(5) *Chłopak kocha dziewczynę / się w dziewczynie. (A boy loves a girl.)*
(6) *Córka niepokoi matkę. (Daughter worries (her) mother.)*
*Matka niepokoi się córką/o córkę. (Mother is worried about (her) daughter.)*
(7) *Żołnierz obronił towarzysza/się przed atakiem.*
*(A soldier defend his comrade/himself from the attack.)*
(8) *Chłopak spotkał dziewczynę / się z dziewczyną. (A boy met a girl.)*
*Chłopak i dziewczyna spotkali się (ze sobą). (A boy and a girl met (each other).)*
(9) *Kelner stłukł szklanki. (A waiter broke glasses.)*
*Szklanki stłukły się. (Glasses broke.)*

Semantically, *reflexive deletion* alternation preserves the number of slots, as *się* plays the role of reflexive pronoun which carries semantic information.

## 5 Rules of Grouping Schemata

One can interpret an alternation as a binary relation between schemata. We will interpret a set-theoretical sum of such relations for verb $v$ as $\mathscr{A}_v$. Such relation is symmetric

[2] These alternations cannot be differentiated at the level of schemata.

and it can be thought of as being reflexive. Nevertheless, it is not transitive. However, if we limit ourselves to alternations preserving the number of arguments, the corresponding relation $\mathscr{A}_v^0$ can be extended to the equivalence relation $\mathscr{A}_v^{\equiv}$, which imposes the partition into equivalence classes on $G_v$.

Unfortunately, this is not the case for alternations "losing" slots. The reason for this is that if all slots are preserved, we can find a counterpart of each slot in the alternating schema via the path of alternations. However, if a particular slot is absent, the path is cut.

Consider the following examples of deletion alternation. For the verb *zakończyć* (*to finish*) the shorter schema is common for two cases of alternation, whereas for the verb *postawić* (polysemous) it is the longer one.[3]

(10) a. *Drużyna zakończyła sezon* (*sukcesem*).
zakończyć np:acc (np:inst) np:nom
(*The team finished the season with success.*)

b. *Drużyna zakończyła sezon* (*bez sukcesu*).
zakończyć np:acc np:nom (prepnp:bez:gen)
(*The team finished the season without success.*)

(11) a. *Gospodarz postawił butelkę wina* (*na stół*).
postawić np:acc np:nom (prepnp:na:acc)
(*The host put a bottle of wine on the table.*)

b. *Gracz postawił* (*dużą sumę*) *na swojego faworyta.*
postawić (np:acc) np:nom prepnp:na:acc
(*The gambler bet* (*a big sum*) *on his favourite.*)

On the basis of such observations, we came to the conclusion that linking schemata representing the same meaning of a verb is much more probable in the case of losing arguments alternations sharing the shorter of the schemata than in the case of alternations sharing the longer one. This results in a procedure of grouping verb schemata consisting of four steps. Let $g^s$, $g^l$ be the shorter and the longer schema of a pair related by a losing slot alternation, respectively.

A. Grouping schemata related by alternations preserving all slots;
B. Adding $g^s$ to a group containing $g^l$;
C. Subsuming a group containing $g^s$ by a group containing $g^l$;
D. Joining groups containing $g^s$ and some $g^l$, $g^{l\prime}$ having consistent slots.

After each step the number of groups of schemata decreases, whereas their size increases, hence we obtain a chain w.r.t. inclusion relation $\subseteq$. If the 3rd and 4th steps are performed, the 2nd step is redundant. The result of the 1st step forms a partition of the set of schemata $G_v$, the results of the 2nd and 3rd step are not a partition, unless verb $v$ does not participate in any losing slot alternation. The result of the 4th step could be a partition, but it does not need to. The consistency of slots is checked on the basis of information about their obliqueness hierarchy, cf. [10].

[3] Examples are supplied with corresponding schemata, not with glosses.

# 6 Experiments

The experiments were performed using the manually prepared set of alternations MAN and the automatically obtained set of alternations AUTO [9]. Alternations, in which verb *rozpocząć* (*to begin*) participates, are presented in (12). Semantically consistent arguments participating in the alternation are displayed as **np:nom**, and semantically dropped are displayed as np:nom. All schemata of verb *rozpocząć* are semantically related and form a single group. Groups of schemata of verb *minąć* are shown in (13).

(12)

| | | |
|---|---|---|
| (12) | reflexive shift | |
| | **np:acc** np:inst np:nom | np:inst **np:nom** sie |
| | **np:acc** np:nom | **np:nom** sie |
| | **np:acc** np:nom prepnp:od:gen | **np:nom** prepnp:od:gen sie |
| | simple | |
| | np:acc **np:inst** np:nom | np:acc np:nom **prepnp:od:gen** |
| | **np:inst** np:nom sie | np:nom **prepnp:od:gen** sie |
| | deletion | |
| | np:acc np:inst np:nom | np:acc np:nom |
| | np:acc np:nom prepnp:od:gen | np:acc np:nom |
| | np:inst np:nom sie | np:nom sie |
| | np:nom prepnp:dla:gen sie | np:nom sie |
| | np:nom prepnp:od:gen sie | np:nom sie |
| | reflexive cross | |
| | **np:nom prepnp:dla:gen** sie | **np:acc np:nom** |

| | | | | |
|---|---|---|---|---|
| (13) | minąć | *to pass* | minąć | *to pass* |
| | | np:dat np:nom | | np:acc np:nom |
| | | np:nom | | np:nom prepnp:z:inst sie |
| | | np:nom prepnp:z:inst | | np:nom sie |
| | minąć | *to pass* | minąć | *to differ* |
| | | np:nom | | np:nom prepnp:z:inst sie |
| | | np:nom prepnp:od:gen | | |
| | | np:nom prepnp:z:inst | | |

## 6.1 Validation

Three popular clustering validation methods exist, based on the co-occurrence of two elements (simple frames) in two partitions of a particular data set. Let

- $b$ be the number of pairs co-occurring in both sets,
- $c$ be the number of pairs co-occurring only in the validated set,
- $g$ be the number of pairs co-occurring only in the gold standard,
- $n$ be the number of pairs co-occurring in neither of sets.

Then *Rand statistics* (R), *Jaccard coefficient* (J) and *Folkes and Mallows index* (FM) are given by the equations [11]:

$$R = \frac{b+n}{b+c+g+n}, \qquad J = \frac{b}{b+c+g}, \qquad FM = \frac{b}{\sqrt{b+c}\sqrt{b+g}}.$$

Rand statistics resemble in a way the accuracy measure used in typical lexical acquisition tasks. With such a point of view, Jaccard Coefficient and Folkes and Mallows index could be interpreted as counterparts of combinations of precision and recall.

In order to apply these to our data, we need to remember about the specificity of the problem of grouping syntactic schemata. First, instead of one big set of data we have several verbs, and their schemata are grouped separately. Their validation may be calculated cumulatively or on average. Second, groups are not disjoint. One group can even include the other. In particular, this concerns even single-element groups. Because of that, we consider elements of such singletons as special pairs counted into $b, c$ or $g$, respectively.

The results of the validation are presented in table 1. For experiments performed on the manually prepared set of alternations, all indices increase while groups are aggregated, with a slight decrease between steps B and C. After each step $b$ and $c$ increase whereas $g$ and $n$ decrease ($b + c$ and $b + g$ are constant). A bad result for A is a consequence of the fact that there is a proportionally large amount of singleton groups obtained in this way. The more thorough examination of results shows the strongest increase of $c$ during step C. Surprisingly, step A performs best for automatically obtained data AUTO. However, for other steps manual data MAN are the best. For AUTO, the best results are obtained for step B. This means that aggregating data further results in aggregating errors.

**Table 1.** Validation of grouping schemata

| data/ | | cumulatively | | | average | | |
|---|---|---|---|---|---|---|---|
| method | | R | J | FM | R | J | FM |
| MAN | A | 32.2 | 10.1 | 25.0 | 43.0 | 20.6 | 34.2 |
| | B | 59.1 | 46.4 | 66.8 | 72.5 | 64.0 | 77.3 |
| | C | 57.7 | 48.6 | 66.8 | 70.8 | 63.6 | 76.2 |
| | D | 63.6 | 57.9 | 73.5 | 73.2 | 67.9 | 80.0 |
| AUTO | A | 34.4 | 19.4 | 36.7 | 41.7 | 25.3 | 39.7 |
| | B | 53.1 | 45.8 | 63.7 | 62.3 | 53.8 | 70.1 |
| | C | 50.0 | 43.4 | 61.3 | 60.3 | 53.0 | 68.5 |
| | D | 51.0 | 45.7 | 56.3 | 59.2 | 53.7 | 69.0 |

## 7 Conclusions

In this paper a heuristics for grouping schemata based on their participation in diathesis alternations has been presented. Obtained groups represent semantic valence dictionary entries forming particular meanings of a verb. Four methods consisting in a gradual aggregation of groups of schemata of a verb participating in any alternation has been proposed. Poor validation results for the simplest method involving only alternations preserving the number of arguments confirms that alternations changing the number of arguments relate schemata semantically. However, if we want to have most pure groups, we should choose the method ending with step B, whereas if one wishes mostly connected groups, we should perform the whole process. Ending with step C is never optimal.

The method is sensitive to the quality of data. For automatically established alternations, we obtain worse groups of schemata, and advanced aggregation of groups lowers their quality further.

The experiments should be performed on larger sets of verbs. The method could be applied to other languages as well.

**Acknowledgements.** This research is supported by the POIG.01.01.02-14-013/09 project which is co-financed by the European Union under the European Regional Development Fund.

## References

1. Agirre, E., Martinez, D.: Learning class-to-class selectional preferences. In: Proceedings of the 5th CoNNL-2001 Conference, Toulouse, France, pp. 15–22 (2001)
2. Fillmore, C.J., Johnson, C.R., Petruck, M.R.L.: Background to FrameNet. International Journal of Lexicography 16(3), 235–250 (2003)
3. Gildea, D.J.: Probabilistic model of verb-argument structure. In: Proceedings of the 6th CoNNL-2002 Conference, Taipei, Taiwan, pp. 308–314 (2002)
4. Hajič, J.: Complex corpus annotation: The Prague dependency treebank. In: Šimková, M. (ed.) Insight into Slovak and Czech Corpus Linguistics, Veda, Bratislava, Slovakia, pp. 54–73 (2005)
5. Hajnicz, E.: Semantic annotation of verb arguments in shallow parsed Polish sentences by means of EM selection algorithm. In: Marciniak, M., Mykowiecka, A. (eds.) Aspects of Natural Language Processing. LNCS, vol. 5070, pp. 211–240. Springer, Heidelberg (2009)
6. Hajnicz, E.: Generalizing the EM-based semantic category annotation of NP/PP heads to wordnet synsets. In: Vetulani, Z. (ed.) Proceedings of the 4th L&TC Conference, Poznań Poland, pp. 432–436 (2009)
7. Hajnicz, E.: Problems with pruning in automatic creation of semantic valence dictionary for Polish. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 131–138. Springer, Heidelberg (2009)
8. Hajnicz, E.: Aggregating entries of semantic valence dictionary of Polish verbs. In: Bertinetto, P.M., Korhonen, A., Lenci, A., Melinger, A., Schulte im Walde, S., Villavicencio, A. (eds.) Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features (Verb 2010), Pisa, Italy, Scuola Normale Superiore and Università di Pisa, pp. 49–54 (2010)
9. Hajnicz, E.: Similarity-based method of detecting diathesis alternations in semantic valence dictionary of Polish verbs. In: International Joint Conference on Security and Intelligent Information Systems, Warsaw, Poland (2011)
10. Hajnicz, E.: Ordering slots of semantically related schemata of Polish verbs. In preparation
11. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems 17(2/3), 107–145 (2001)
12. Hlaváčková, D., Horák, A.: VerbaLex — new comprehensive lexicon of verb valences for Czech. In: Proceedings of the Third International Seminar on Computer Treatment of Slavic and East European Languages, Bratislava, Slovakia, pp. 107–115 (2006)
13. Joanis, E., Stevenson, S., James, D.: A general feature space for automatic verb classification. Natural Language Engineering 14(3), 337–367 (2008)
14. Kipper-Schuler, K.: VerbNet: A broad coverage, comprehensive verb lexicon. PhD thesis, Computer and Information Science Department, University of Pennsylvania (2005)

15. Lapata, M.: Acquiring lexical generalizations from corpora: a case study for diathesis alternations. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 1999), College Park, MA, pp. 397–404 (1999)
16. Lapata, M., Brew, C.: Verb class disambiguation using informative priors. Computational Linguistics 30(1), 45–73 (2004)
17. Levin, B.: English verb classes and alternation: a preliminary investigation. University of Chicago Press, Chicago (1993)
18. Li, H., Abe, N.: Generalizing case frames using a thesaurus and the MDL principle. Computational Linguistics 24(2), 217–244 (1998)
19. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313–330 (1993)
20. McCarthy, D.: Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences. PhD thesis, University of Sussex (2001)
21. Merlo, P., Stevenson, S.: Automatic verb classification based on statistical distributions of argument structure. Computational Linguistics 27(3), 373–408 (2001)
22. Pala, K., Smrž, P.: Building the Czech WordNet. Romanian Journal of Information Science and Technology 7(2–3), 79–88 (2004)
23. Palmer, M., Kingsbury, P., Gildea, D.J.: The proposition bank: an annotated corpus of semantic roles. Computational Linguistics 31(1), 71–106 (2005)
24. Resnik, P.: Selectional constrains: An-information-theoretic model and its computational realization. Cognition 61, 127–159 (1996)
25. Schulte im Walde, S.: Clustering verbs semantically according to their alternation behaviour. In: Proceedings of the COLING-2000 Conference, Saarbrücken, Germany, pp. 747–753 (2000)
26. Szupryczyńska, M.: Syntaktyczna klasyfikacja czasowników przybiernikowych. Państwowe Wydawnictwo Naukowe, Poznań, Poland (1973)
27. Świdziński, M.: Syntactic Dictionary of Polish Verbs. Uniwersytet Warszawski / Universiteit van Amsterdam (1994)
28. Žabokrtský, Z., Lopatková, M.: Valency information in VALLEX 2.0: Logical structure of the lexicon. The Prague Bulletin of Mathematical Linguistics 87, 41–60 (2007)

# Hierarchical Dialogue System for Guide Robot in Shopping Mall Environments

Maria Prischepa and Victor Budkov

St. Petersburg Institute for Informatics and Automation of RAS
SPIIRAS, 39, 14th line, St. Petersburg, Russia
{ronzhin,karpov,kipytkova}@iias.spb.su
www.spiiras.nw.ru/speech

**Abstract.** Specifics of dialogue model and designing multimodal user interface for a mobile robot intended for inquiry and navigation services for visitors in a shopping mall are considered in the paper. Humanoid shape and style of communication became more popular and possible for state-of-the-art social robot due to rapid development of the embedded systems and the methods for audio-visual processing of natural modalities. The proposed user interface is based on distant speech recognition and talking head techniques. The hierarchical structure of sub-dialogues connected with robot function modes is proposed to reduce the complexity of speech processing. The template-based models for key phrases were prepared in each sub-dialogue for promotion, inquiry, and guide modes of the mobile robot.

**Keywords:** Spoken dialogue system, dialogue management, multimodal interfaces, talking head, speaker localization.

## 1 Introduction

Fundamental principles of the field of human-computer interaction lays the basis for the design of dialogue models of user interaction with robots, also the capabilities of modern hardware and software that implement the input, output and processing of information channels available to the user are taken into account. Modern user interfaces can be divided into the two basic categories: standard graphic and multimodal interfaces. The standard graphical user interface (GUI) remained the most common before the appearance of complex robotic systems of mass services. With the development of socially oriented robots, it became clear that the interfaces for interaction of the robot with the human should be simpler, more intuitive and do not require additional knowledge and training. Therefore, the development of social robots was jointly conducted with designing of multimodal interfaces. Figure 1 presents some variants of user interfaces, which are used in human-machine interaction area.

The standard interface is a graphical user menu, which includes information inputting by a user in manual mode (keyboard, mouse, touchscreen monitor). The most widespread of such interface has received in a self-service machine, such as payment terminals or ATM services. This kind of interaction is not always convenient for a user, and often even impossible, for example, people with disabilities are not able to interact

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 163–170, 2011.

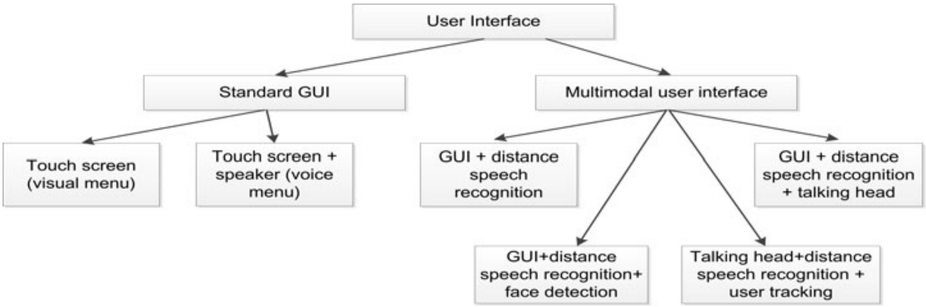

User Interface
Standard GUI
Multimodal user interface
Touch screen (visual menu)
Touch screen + speaker (voice menu)
GUI + distance speech recognition
GUI + distance speech recognition + talking head
GUI+distance speech recognition+ face detection
Talking head+distance speech recognition + user tracking

**Fig. 1.** User interfaces classification

in this way (blind, armless, etc.). To increase the opportunities of graphical user interface voice prompts to the menu should be used in self-service machines and robots are used.

For example, a robot Neel, developed by an Indian group HitechRoboticSystemz Ltd, is an autonomous reference robot, which provides information services to visitors in shopping mall [1]. The robot navigation system is based on laser sensors and route planning for a given map. The robot is equipped with a touch screen with graphical menus, menu items can be synthesized by Microsoft Windows TTS. The system of interaction with a user applies speech synthesis and a graphical menu. A user selects goods or services on the touch screen, the robot pronounces his/her choice and the response to the user's query. Neel robot is connected to the information network of the shopping center and notified of all changes, availability of goods and services. Also, when interacting with people the robot creates a database of visitors and their preferences based on analysis of user queries. Currently, the robot is able to independently navigate a given route and to identify obstacles. The user interface is based on JavaFX, which allows quick change of graphical part of the interface.

Much greater attention is now given to the development of queuing systems with multimodal user interfaces based on analysis of speech, gestures, and graphical user interface, three-dimensional model of a human head with a strong articulation of speech, facial expressions and other natural means of communication for interpersonal communication. Figure 1 shows only four choices of modality combinations, most commonly used in the development of application systems of human-robot interaction.

System with a multimodal user interface, including at least speech recognition and synthesis, in addition to the graphical menu, will benefit for a lot more groups. For example, visually impaired people can interact with the system in a natural way using speech. An example of such systems is a robot FriDA, which was developed by Korean company DASA TECH Co. Ltd. FriDA. This robot is equipped with a touch-screen monitor, speakers, and a microphone array [2]. The monitor has standard graphic menus, as well as speakers and microphones to ensure system of synthesis and speech recognition. The robot is designed to provide reference information at the airport in a verbal dialogue mode and can display and pronounce data required by user.

Systems with three-dimensional avatar of the human head are able to communicate with hearing disabled people. Lip movements of avatars are synchronized with the speech signal, which makes possibility for lip reading. For example, a robot secretary HALA, developed at the University of Carnegie Mellon, is equipped with a touch screen, which displays animated avatars, speaker, microphone and an infrared sensor to determine the presence of a user [3]. HALA can lead voice dialogue with a user in Arabic and English languages, the avatar is used for verbal expressions (movements of the lips is applied in the process of speech synthesis) and nonverbal means (shaking his head, facial movements).

Recently there was a tendency to create humanoid robots with the approximate shape of the hull, with varying degrees, to the human body shape. Such robots are able to interact with a person, not only through speech but also with gestures. Typically, these robots are not equipped with monitors, therefore they have not any graphical interface. For example, the robot Robotinho, developed at the University of Bonn in Germany, has a humanoid form, and can interact with humans through speech, gestures and facial expressions [4]. The robot uses mixed system of dialogue, and is able to determine position of a user and his face, as well as to recognize and synthesize speech. Robotinho can express its emotional state and communicate with many people simultaneously. Since the robot has a humanoid body shape, it can nonverbally communicate with users through gestures during the dialogue, as well as attract users' attention to itself or to the objects of the environment by gestures or gaze direction. The robot detects a user with two laser range finders, and then he finds a human face with two video cameras. When interacting with users it creates a database of users containing user's face images and his/her preferences, based on the query history. In future the robot will be able to identify user.

Thus, the appointment of the robot and the possibilities of potential users are necessary to consider at the development of multimodal interfaces for a social robot. Ways of interaction must be easy-to-use and do not require special training of users. Speech and multimodal interfaces with speech processing, are being actively researched and applied in robotic systems although. Despite the fact that user interaction with social robots in most cases takes place in an environment with high noises, speech interfaces, and multimodal, including speech processing, are being actively studied and applied research in robotic systems [5,6,7].

## 2 Hierarchical Dialogue Model Based on Robot Function Modes

The developed mobile information robot provides reference services to visitors of shopping malls. Therefore, the interaction dialog model corresponds to robot's tasks. Figure 2 shows the robot main modes: (1) promotion (2) inquiry, (3) guide, (4) moving to the base. A graph structure which consists of characteristics of each mode and corresponding sub dialog between users and the robot.

In the ”Inquiry” mode data required to a user are displayed on the screen and pronounced by speech synthesis system. For example, the output of the current location of the user and the robot, the route to the point of interesting (POI), searching for goods and services on the database shopping center, search the store by the name or belonging to a category of goods are carried out.

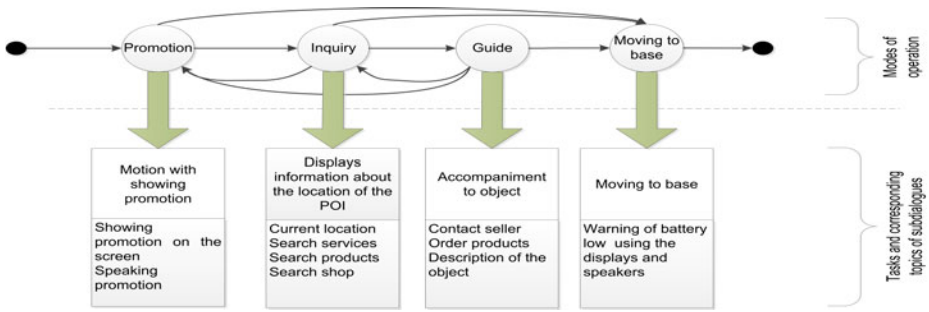

Promotion
Inquiry
Guide
Moving to base
Modes of operation
Motion with showing promotion
Showing promotion on the screen
Speaking promotion
Displays information about the location of the POI
Current location
Search services
Search products
Search shop
Accompaniment to object
Contact seller
Order products
Description of the object
Moving to base
Warning of battery low using the displays and speakers
Tasks and corresponding topics of subdialogues

**Fig. 2.** The main tasks of the robot and corresponding sub dialogues

The ”Guide” mode provides not only reference services for determination of location of objects (shops), which are interested to a user, but also the possibility of his/her tracking there. While escorting user the robot can display information about the object, make online ordering of the goods, or make direct contact with a representative of the store.

The information about current promotions in shops, goods and service is displayed on screen and voiced in the ”Promotion” mode by the talking head. In the ”Moving to base” mode message that the robot is out of order is displayed on the screen. If the robot will be blocked from every side the talking head will give voice request for free way. At the same time a message about critical situation is sent to the operator. When the robot battery has low charge it can change the current mode to ”Moving to base”, but first of all the robot will give information about changing work mode to the current user.

The initial state of the robot is the ”Promotion” mode. When the robot detects a visitor in the area of video observation it turns to the ”Inquiry” mode, which primarily runs greeting sub dialog and suggests to select a category of the services. After the choosing and the finding certain goods or store, the result is displayed on the screen and voiced by the talking head. Then, the robot proposes to accompany the visitor to the selected object. In case of visitor’s positive answer, the robot switches to ”Guide” mode. In other case the robot proposes to choose something else. If the user declines, then the robot comes back to the ”Promotion” mode.

One of the main issues in designing a dialogue strategy is the degree of initiative let to the user in each dialogue state [8]: (1) system-led: the system has the only initiative and asks sequences of precise questions to the user. The user is then supposed to answer those questions and provide only the information he/she has been asked for; (2) user-led: the user has the only initiative and asks for information to the system. The system is supposed to interpret correctly the user’s query and to answer those precise questions without asking for more details; (3) mixed initiative: the user and the system share the control to cooperate in order to achieve the user’s goal.

The use of the dialogue model in the shopping center was selected because the system-led mode in most cases degrades the naturalness of interaction and imposes strong restrictions on the user’s phrase. On the other hand, the modern means of natural language processing have not yet reached the level of completely user-led strategy.

In addition, the system should be available to people with varying degrees of training. Also visitors, by psychological factors are afraid to start a dialogue with the robot. The user-led strategy means that a user has a freedom to build a course of dialogue and every phrase. This leads to a significant increase in databases, which requires a speech recognition system to handle voice messages. There is a tradeoff between making the robot's recognition grammars sufficiently large so that people can express themselves somewhat freely versus making the grammars small enough so that the system runs with high accuracy and in real-time[9]. We tuned speech recognition performance by creating the sets of the specialized grammars for different robot contexts and the robot's software dynamically activates a corresponding subset of grammars depending on the context of the robot activity. This allows us to overcome the combined increase in parsing time due to incorporating natural syntactic variability in the recognition grammars.

Instead of keeping a single enormous recognition grammar active, the robot keeps subsets of small grammars active in parallel, given what it currently expects to hear. The key assumptions here are that certain types of utterances are only likely to be said under particular circumstances, and these are circumstances among which the robot is capable of distinguishing[9].

Structuring the dialogue model in accordance with the current mode of operation is shown in Figure 3. Such division has several advantages. First, the system performance is increased due to the fact that each sub dialog has own vocabulary, so the search is performed in the reduced area. Second, it simplifies the procedure of adding new or updating existing sub dialogs. There is no reason for redo of the entire system, that's enough to add a new sub dialog and conditions of its call.

Let us consider structure of sub dialogs and transitions between them. In the beginning the system greets user and suggests to listen a brief information about promotions. If a user disagrees then the robot suggests user to ask information about shops or goods. In other case objects which held in the current promotion, are marked at the screen and the robot changes the sub dialog to the ”Information about promotions”, in which data on all the objects and their events are provided. A visitor can select interesting information by clicking on the appropriate store displayed on the map and pronouncing its name, and if user wants the robot changes sub dialogue to ”Escort” and escorts him/her to the selected point. After displaying information about all actions or a specific sale in the store, the robot asks the user about necessity of information about other shops, goods and etc. If the user refuses to continue the dialogue, the robot turns to the sub dialog ”Completing the dialogue”. If user agrees then robot selects sub dialog ”Information about shops” in which user can ask to find certain shop, good or a set of shops that sell certain goods category. When a shop is selected, the robot also asks to escort user to it.

Each sub dialog additionallyto the common vocabulary, which is used in all modes, also has own dictionary and a set of phrases different from the other sub dialogs. The customer profile is generated in process of interaction between the robot and him/her, which includes the main personal data and the preferences in choosing goods or shops. Figure 4 presents a structure of available phrases, which were compiled like a grammar, because in developed model for dialog with customer the main aim is determination of POI name or goods name. The phrase can include the name of POI (set of elements

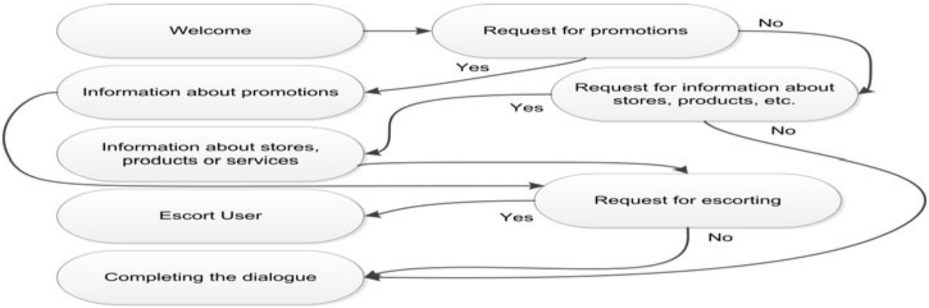

Welcome
Request for promotions
No
Yes
Information about promotions
Request for information about stores, products, etc.
Yes
No
Information about stores, products or services
Request for escorting
Escort User
Yes
No
Completing the dialogue

**Fig. 3.** Structure of sub dialogs

$shop_list, $cafe_list, $service_list); goods name (set of elements $goods_list) or name with other words "shop", "cafe" as well as introduction phrase with verb "buy", "find" and other (set of elements $where_buy, $where_find).

If a user phrase contains POI name, so the robot calculates a path to the entrance to the interesting POI. If a user chooses some goods/services, the robot searches the objects, where it is sold, after that list of all accepted objects are shown on the screen, where the customer can choose one of them. After that robot changes the work mode to "Guide". The accumulated information on movement and interaction with users, itinerary, completed tasks as well as visitor's preference are archived for future optimization of the robot's control system.

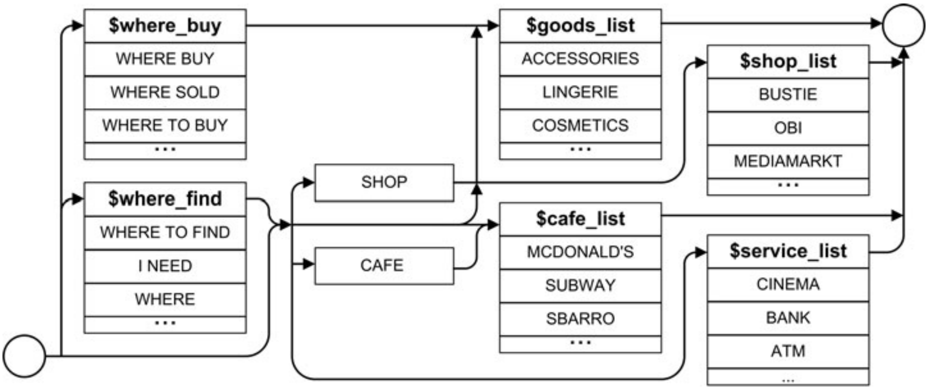


**Fig. 4.** The template-based grammar model for phrase recognition

The example of a dialog, which is used during user-robot interaction and described by the proposed template based model for speech recognition and synthesis, is presented below (R - robot, U - user): R: Hello, Would you like to know about special promotions in the mall?

U: Yes, I want.

R: Today the shops [$shop_list_1] and [$shop_list_2] have special promotions. Would you like to know more about one of it?

U: Yes.

R: Please, could you specify your request?

U: What kind of special promotions occurs in shop [$shop_list_2]?

R: In the shop [$shop_list_2] there is discount on [$goods_list]. [Text of the promotions]. Would you like to know anything else about special promotions?

U: No.

R: Would you like to know about goods, shops and services in the shopping mall?

U: Yes, where's a [$goods_list] shop?

R: [$goods_list] shops are presented on the screen. Please, choose ashopyou are interested in.

U: [$shop_list].

R: The shop [$shop_list] is presented on the map. Do you need escorting?

U: No.

R: Would you like to know anything else?

U: No.

R: Thank you for using our system. Goodbye.

As seen from the dialogue example the shops and the goods can be found both by name and by category. When a user asks a product category the system searches for a shop, which sells that. In the developed dialogue model the main objective is to identify the name of the store or product. User's phrase can contain only the name of a store (the elements of [$shop_list]), the name of a product (the elements of [$goods_list]) or a name with the additional words ”shop”,”cafes”, as well as introductory turnovers with the verbs ”buy”, ”find” and others.

## 3 Conclusion

Intuitive user interfaces is a key to successful human-computer interaction. Common interfaces that represent various graphical menus, do not satisfy/ meet all needs, and can not be used by some groups of people. Therefore, during the development of service robots some attention should be paid to multimodal user interfaces, which provide a natural interaction for a beginner. The main advantage of the developed robot interaction model with visitors in shopping malls is its modularity. The existence of separate dictionaries for each sub dialog enhances the quality of speech recognition and simplifies the modification and completion of the entire interactive model. Further research will be focused on development of the sub dialogs, as well as the creation of new based on data obtained from testing the robot in real conditions.

**Acknowledgments.** This work is supported by the Russian Federal Targeted Program (contract No.876) of the Ministry of Science and Education of Russia and by the Russian Foundation for Basic Research (project No. 10-08-00199-a).

## References

1. Datta, C., Kapuria, A., Vijay, R.: A pilot study to understand requirements of a shopping mall robot. In: Proceedings of HRI 2011, pp. 127–128 (2011)
2. http://int.dasarobot.com/ucp/pages/product/information-robots/frida/
3. Makatchev, M., Fanaswala, I., Abdulsalam, A., Browning, B., Ghazzawi, W., Sakr, M., Simmons, R.: Dialogue Patterns of an Arabic Robot Receptionist. In: Proceedings of HRI 2010, pp. 167–168 (2010)
4. Nieuwenhuisen, M., Stuckler, J., Behnke, S.: Intuitive Multimodal Interaction for Service Robots. In: Proceedings of HRI 2010, pp. 177–178 (2010)
5. Kollar, T., Tellex, S., Roy, D., Roy, N.: Toward Understanding Natural Language Directions. In: Proceedings of HRI 2010, pp. 259-266 (2010)
6. Cantrell, R., Scheutz, M., Schermerhorn, P., Wu, X.: Robust Spoken Instruction Understanding for HRI. In: Proceedings of HRI 2010, pp. 275-282 (2010)
7. Kriz, S., Anderson, G., Trafton, G.: Robot-Directed Speech: Using Language to Assess First-Time Users' Conceptualizations of a Robot. In: Proceedings of HRI 2010, pp. 267-274 (2010)
8. Pietquin, O.: A framework for unsupervised learning of dialogue strategies, p. 246. UCL presses, London (2004)
9. Coen, M.H.: Design principles for intelligent environments. In: Proc. of the 15 National Conference on Artificial intelligence, pp. 547–554 (1998)

# Identifying Concatenation Discontinuities by Hierarchical Divisive Clustering of Pitch Contours*

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.** In this paper, we present the results of a clustering experiment, the aim of which was to show whether or not the proximity of pitch contours is sufficient condition for perceptually smooth transitions at concatenation points in concatenative speech synthesis. The experiment was motivated by a previous finding which had shown that the support vector machine (SVM) classifiers are capable of separating with a high accuracy perceptually continuous and discontinuous joins using the pitch contours extracted from the vicinity of concatenation points as predictors. The experiment has shown that clustering of observations in a form of pitch contours represented in different scales using the euclidean distance as a metric does not prove to be a reliable way of identifying discontinuities at concatenation points.

**Keywords:** speech synthesis, unit selection, concatenation cost, pitch contours, hierarchical divisive clustering.

## 1 Introduction

Despite the increasing popularity of HMM based speech synthesis methods, the unit selection concatenative systems still represent the mainstream in many practical applications, especially in limited domains where synthesized chunks are combined with pre-recorded prompts. In such applications, the ability of the unit selection to deliver highly natural and to the recordings well fitting output are the key factors. Not surprisingly, the unit selection also remains the first option for eBook reading applications, which have been acquiring a lot of interest over recent years.

Among the unit selection related issues that continue to be non-resolved, the audible discontinuities appearing at concatenation points play an important role. According to the original idea [1], the amount of discontinuity introduced by concatenating successive units should be reflected by a *concatenation (join) cost function*. Since phase, pitch and spectral envelope mismatches are believed to be the main sources of the discontinuities [2], ideal concatenation cost function should cover all these aspects.

* This research was supported by the Grant Agency of the Czech Republic, project No. GAČR 102/09/0989; by the Technology Agency of the Czech Republic, project No. TA01030476. The work has also been supported by the grant of the University of West Bohemia, project No. SGS-2010-054.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 171–178, 2011.

Many studies have been published over last one and a half decades focusing on the spectral mismatches in the first place while eliminating the other sources of discontinuities [3], [4], [5], to name but a few. Despite the considerable amount of efforts, none of them unfortunately succeeded to provide a clear answer on how to measure the discontinuities at concatenation points. The presented results have even sometimes been in contradiction.

In our previous work [6] dealing with vowels, and also in an informal analysis of concatenation artifacts present in the outputs of our TTS system [7], it was found out that a large number of audible discontinuities tend to appear at joins where units having originally incoherent $F0$ values in the area of the prospective concatenation points are put together. Other possible sources of discontinuities were also identified but not in such an extend.

In line with these observations, we decided to extract pitch contours from the vicinity of concatenation points and use them as predictors in the discontinuity detection task performed by the SVM classifiers. This experiment has shown that the $F0$ contours contain enough information for separating continuous and discontinuous concatenations with a high accuracy reaching the range around 90%.

This finding has lead us to the idea of conducting a clustering experiment having observations in a form of $F0$ contours, and using the euclidean distance as a metric. If the clustering proved to be a feasible way of identifying units that can be well concatenated, it would be more easily implementable into our unit selection based TTS system [7] than the knowledge learned by the SVM classifiers.

In our experiment, we used perceptual data collected in two listening tests designed for a male and a female voice. The collection of perceptual data is described in Sec. 2. The sets of observations that were to be clustered are defined in Sec. 3, and in Sec. 4, we draw the experiment conclusions and outline our future work intentions.

## 2 Perceptual Data Collection

In order to collect data that can be used for the evaluation and design of the concatenation cost functions, we had conducted two listening tests in the past—one to collect male voice data, one for female voice data. In the following subsections, the content and the evaluation procedure of these listening tests are briefly described. More details may be found in [8] and [9].

### 2.1 Test Material

Recordings covering five Czech short vowels in all consonantal contexts were made in an anechoic room by two professional speakers—male and female. The recorded scripts were composed of three word sentences containing consonant-vowel-consonant (CVC) word in the middle each, e.g. `/kra:lofski:` **`kat`** `konal/` (Czech SAMPA notation). Recorded data were re-synthesized using the “half sentence” method [6]. This method consists in cutting the sentences in the middle of the vowels in the central words and combining the left and right parts, which results in a large set of sentences containing only one concatenation point in the middle of the central CVC word each. Note that the concatenations were done pitch synchronously to avoid phase mismatches, but no smoothing algorithm was applied.

Since the whole set of synthesized sentences was too large to be entirely used as listening test stimuli, and we did not want to make a random selection, different concatenation cost functions were applied to collect a limited set of sentences, which were then included to the listening tests stimuli. The selection was done with the expectation to obtain slightly larger number of discontinuous ratings in the listening tests. The total number of sentences presented to the listeners in each listening test was 1310, including some natural and revision sentences.

### 2.2 Listening Tests Subjects

The subjects were university students, all native speakers of Czech. A few 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 two tests. All subjects were paid upon completion of the tests.

### 2.3 Listening Tests Procedure

The task of the listeners was to assess the concatenations on both the five-point scale (*no join at all*, *unnatural but not disturbing*, *slightly perceived join*, *highly perceived join*, and *highly disturbing join*), and the 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 the synthesized sentences. Note that in the preliminary classification experiment as well as the clustering experiment presented in this paper only the binary scale ratings have been used.

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 a silent environment and using headphones. 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 [8]. To help the listeners calibrate for the more fine grained scale, a preparation phase was included containing various examples of audible discontinuities. It was allowed to listen to the calibration sentences at any time during the listening test. There were no restrictions on how many times listeners played each sentence before assessing it.

### 2.4 Listening Test Evaluation and Results

In order to identify listeners who did not show good agreement with the majority, a rigorous analysis of the listeners' ratings has been performed [8]. We ranked the participants according to the scores obtained by the analysis, and 9 and 6 participants were excluded from the male and female voice listening tests, respectively. The ratings of these listeners were not used to create a set of "facts" and to calculate agreements scores as described below.

As a next step, we have collected two sets of "facts", i.e. sentences that were assessed by more than or equal to 80% of listeners in the same way on the binary scale. The set of "facts" can be formally described as:

$$\mathrm{sent}_i \in \mathrm{FACTS} \quad \Leftrightarrow \quad \frac{\mathrm{N}_i^+}{\mathrm{N}_i} \geq 0.8 \vee \frac{\mathrm{N}_i^-}{\mathrm{N}_i} \geq 0.8, \tag{1}$$

where $\mathrm{sent}_i$ is the $i$-th sentence of the test stimuli, FACTS stands for the set of "facts", $\mathrm{N}_i^+$, $\mathrm{N}_i^-$ are the numbers of continuous (i.e. *not perceived join*) and discontinuous (i.e. *perceived join*) ratings given to the $i$-th sentence, respectively, and $\mathrm{N}_i$ is a total number of ratings given to the $i$-th sentence.

The total numbers of collected "facts", which formed data used in our experiments, were 494 for the male voice and 887 for the female voice. Fig. 1 shows the distribution of the "facts" for each vowel.

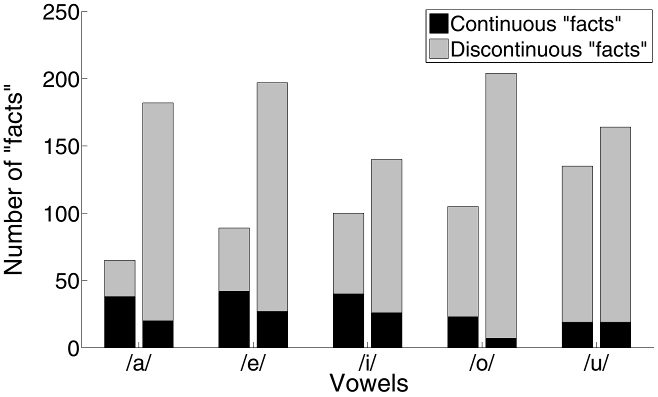


**Fig. 1.** The "facts" collected in the listening tests sorted by the vowels—the left bar in each pair represents the male voice results

In order to have a baseline for the evaluation of results obtained by the SVM classifiers as briefly described in Sec. 3.1, the next step was to calculate an agreement score of each listener using the following formula:

$$\mathrm{AGR\_SCORE}_i = \frac{\mathrm{NUM\_AGR}_i}{\mathrm{FACT\_COUNT}}, \tag{2}$$

where $\mathrm{AGR\_SCORE}_i$ is the agreement score of the $i$-th listener, $\mathrm{NUM\_AGR}_i$ is a number of ratings of the $i$-th listener in agreement with the "fact" rating and FACT_COUNT is the number of collected "facts". The agreement scores of the three least agreeing listeners for each voice are summarized in Tab.1.

**Table 1.** Agreements scores (2) of the three least agreeing listeners participating in the listening tests

| | Male | Female |
|---|---|---|
| `List1` | 0.84 | 0.82 |
| `List2` | 0.87 | 0.83 |
| `List3` | 0.88 | 0.84 |

## 3 Clustering Experiment

### 3.1 Motivation

As already mentioned in Sec. 1, the incentive that lead us to conducting the clustering experiment described in this paper was the very good result of the preliminary classification experiment, in which the SVM based classifiers had shown to perform with the high accuracy reaching the range around 90% the task of separating continuous and discontinuous joins using $F0$ contours extracted from the vicinity of concatenation points as predictors only, not needing information about spectral envelopes or energy. If we look at the listeners' agreement scores summarized in Tab. 1, it can be seen that the classifiers were even overpassing the least agreeing listeners.

Despite obtaining such a good, albeit surprising, result, there have remained non-answered questions. First, to which level the models trained in a very limited prosodic environment, given by the nature of our data, can be generalized for utilization in an unrestricted prosodic domain, i.e. stressed/unstressed syllables, different positions in phrases, different types of phrases, etc. Second, what is the structure that was learned from the data by the SVMs. Natural hypothesis which arose was that it is the proximity of the contours. If this was true, it could be more easily incorporated into our current TTS system than trained SVM models, which would still need to be generalized.

### 3.2 Set of Observations

To maintain equal conditions, the set of observations to be clustered was given by the predictors used in the classification experiment. The only difference was that in addition to the hertz scale, two perceptually motivated scales—mel and semitone—were introduced.

The points of the $F0$ contours were pitch synchronously extracted from the recorded sentences from the areas around prospective concatenation points, and noted as shown in Fig. 2. The set of observations for a given scale can be formally defined as:

$$\mathrm{OBS} = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_M\} \cup \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}\,, \tag{3}$$

where $\mathbf{l}_i = [L_{-4} \ldots L_4]_i$, $\mathbf{p}_j = [P_{-4} \ldots P_4]_j$, and $M$, $N$ are the counts of sentences that appeared in synthesized data as the left and right part, respectively.

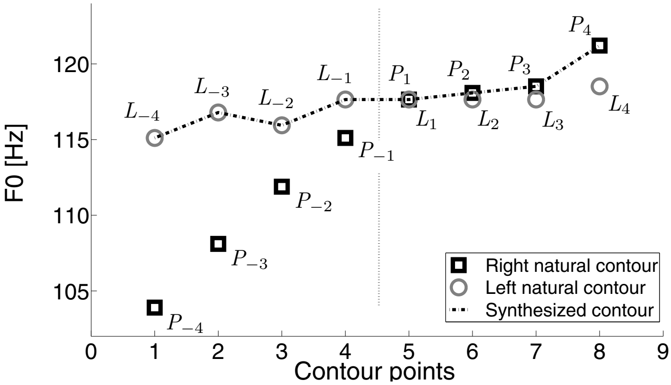


**Fig. 2.** Annotation scheme used for labeling the $F0$ contours. As an example, let $[L_{-4} \dots L_4]$ be the $F0$ contour extracted from the central part of the vowel /a/ in the word /t_Sak/ and $[P_{-4} \dots P_4]$ the contour of /a/ in the word /mas/. Then, the sequence $[L_{-4} \dots L_{-1}, P_1 \dots P_4]$ represents the central part of the concatenated $F0$ contour of the word created as /t_Sa-as/ (in Czech SAMPA notation).

Following the hypothesis formulated in the previous section, if two vectors $\mathbf{l}_i$ and $\mathbf{p}_j$ fall within the same cluster, the resulting $F0$ contour $[L_{-4} \dots L_{-1}, P_1 \dots P_4]$ should be smooth, and the concatenation point not perceived by the listeners.

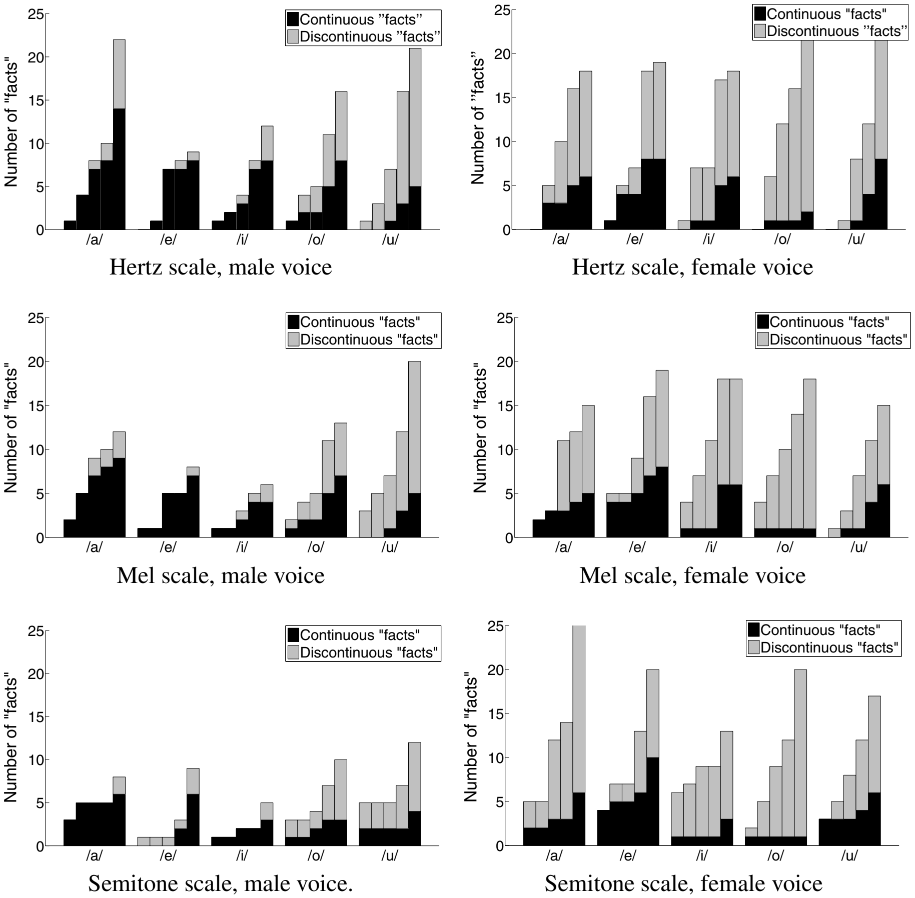
### 3.3 Results

Upon building a clustering tree, different heights (diameters) were applied to cut it into different numbers of clusters. Within each cluster, we then counted the continuous and discontinuous "facts". If our hypothesis was true, the concatenations within small–diameter clusters would be rather continuous, and the larger the diameter would be the larger number of discontinuous concatenations would be appearing. In addition, most of the existing continuous "facts" would be found within clusters.

The results of this procedure are shown in Fig. 3 showing distributions of "facts" within clusters of different diameters for all vowels and comparing different scales. It can be seen from the plots that the hypothesis was not confirmed by the obtained results, especially for the female voice and the male voice vowels /o/ and /u/.

Also for the other male voice vowels, the results were not very convincing taking into account the total amount of continuous "facts" present in our data, and comparing this number with the counts of continuous "facts" present in the clusters. It is obvious from this comparison that there have been many continuous "facts", which did not fall within the same clusters, meaning that the concatenated $F0$ contours have been rather incoherent, but still the listeners did not perceive any discontinuity, and vice versa.

The same unfortunately holds true even for the perceptual scales – mel and semitone – which were expected to lead to better results.



Hertz scale, male voice

Hertz scale, female voice

Mel scale, male voice

Mel scale, female voice

Semitone scale, male voice.

Semitone scale, female voice

**Fig. 3.** Distributions of “facts” sorted by voices and scales. Each particular bar contains distribution for a given cutting height of the clustering tree. For each vowel five clusters’ diameter values were used to plot the distributions, they are ranked in ascending order from left to right. The larger the diameter of final clusters is the more concatenations within clusters appear—including both continuous and discontinuous.

## 4 Conclusions and Future Work

This paper presented results of the clustering experiment. The experiment was motivated by previous finding showing that SVM classifiers are able to separate with the high accuracy perceptually continuous and discontinuous concatenations using $F0$ contours extracted from the vicinity of concatenation points as predictors only, not needing information about spectral envelopes or energy.

Based on this finding, the hypothesis was formulated that the data structure learned by the SVM classifiers is the proximity of the contours. This hypothesis was however disconfirmed by the obtained results as the number of continuous concatenations within clusters was comparatively smaller than their total amount present in our data, and some concatenations within small–diameter clusters were still found to be discontinuous. In addition, a post hoc inspection revealed that there were many concatenations of sentences rated as continuous by the listeners but coming from rather distant clusters. No considerable improvement was found when using perceptual scales—mel and semitone.

Future work will focus on better understanding the knowledge of the $F0$ contours used by the SVM classifiers, and incorporating this knowledge into our TTS system.

## References

1. Hunt, A., Black, A.: Unit selection in a concatenative speech synthesis system using a large speech database. In: ICASSP 1996, Atlanta, Georgia, vol. 1, pp. 373–376 (1996)
2. Dutoit, T.: Corpus–based speech synthesis. In: Benesty, J., Sondhi, M.M., Huang, Y. (eds.) Springer Handbook of Speech Processing. ch. 21, pp. 437–455. Springer, Heidelberg (2008)
3. Klabbers, E., Veldhuis, R.: Reducing audible spectral discontinuities. IEEE Transactions on Speech and Audio Processing 9, 39–51 (2001)
4. Bellegarda, J.R.: A novel discontinuity metric for unit selection text–to–speech synthesis. In: SSW5 2004, Pittsburgh, PA, USA, pp. 133–138 (2004)
5. Vepa, J.: Join cost for unit selection speech synthesis. Ph.D. thesis, University of Edinburgh (2004)
6. 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)
7. 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)
8. Legát, M., Matoušek, J.: Collection and analysis of data for evaluation of concatenation cost functions. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 345–352. Springer, Heidelberg (2010)
9. Legát, M., Matoušek, J.: Analysis of data collected in listening tests for the purpose of evaluation of concatenation cost functions. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS(LNAI), vol. 6836, pp. 33–40. Springer, Heidelberg (2011)

# Identifying Verbal Collocations in Wikipedia Articles*

István Nagy T.[1] and Veronika Vincze[2]

[1] University of Szeged, Department of Informatics,
6720 Szeged, Árpád tér 2., Hungary
[2] MTA-SZTE Research Group on Artificial Intelligence,
6720 Szeged, Tisza Lajos krt. 103., Hungary
{nistvan,vinczev}@inf.u-szeged.hu

**Abstract.** In this paper, we focus on various methods for detecting verbal collocations, i.e. verb-particle constructions and light verb constructions in Wikipedia articles. Our results suggest that for verb-particle constructions, POS-tagging and restriction on the particle seem to yield the best result whereas the combination of POS-tagging, syntactic information and restrictions on the nominal and verbal component have the most beneficial effect on identifying light verb constructions. The identification of multiword semantic units can be successfully exploited in several applications in the fields of machine translation or information extraction.

**Keywords:** multiword expressions, verbal collocations, light verb constructions, verb-particle constructions, Wikipedia.

## 1 Introduction

In natural language processing, the proper treatment of multiword expressions (MWEs) is essential for many higher-level applications (e.g. information extraction or machine translation). Multiword expressions are lexical items that can be decomposed into single words and display idiosyncratic features [11]. To put it differently, they are lexical items that contain space or 'idiosyncratic interpretations that cross word boundaries'. They are frequent in language use and because of their unique and idiosyncratic behavior, they often pose a problem to NLP systems.

In this work, we focus on various methods for detecting verbal collocations, i.e. verb-particle constructions (VPCs) and light verb constructions (LVCs) in Wikipedia articles. First, we offer a short description on characteristic features of these two types of multiword expressions, then related work is presented. Our methods are later described and results achieved are presented. The paper concludes with a discussion of results and future work.

* This work was supported in part by the National Innovation Office of the Hungarian government within the framework of the projects BELAMI and MASZEKER.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 179–186, 2011.

## 2 The Characteristics of Verb-Particle Constructions and Light Verb Constructions

Verb-particle constructions contain a verb and a particle (usually a preposition), e.g. *kick off* or *set out*. They are also called phrasal or prepositional verbs and are highly characteristic of the English language, thus, they occur frequently in texts. The particle modifies the meaning of the verb: it may add aspectual information, may refer to motion or location or may totally change the meaning of the expression.

Light verb constructions are verb and noun combinations in which the verb has lost its meaning to some degree and the noun is used in one of its original senses (e.g. *have a walk* or *give advice*). They are usually distinguished from productive or literal `verb + noun` constructions on the one hand and idiomatic `verb + noun` expressions on the other hand in NLP literature: e.g. Fazly and Stevenson [7] use statistical measures in order to classify subtypes of verb + noun combinations and Diab and Bhutada [6] developed a chunking method for classifying multiword expressions.

Verbal collocations deserve special attention in NLP applications for several reasons. First, their meaning cannot be computed on the basis of the meanings of the parts of the collocation and the way they are related to each other (lack of total compositionality). Thus, the result of translating their parts literally can hardly be considered as the proper translation of the original expression. Second, light verb constructions (e.g. *make a mistake*) often share their syntactic pattern with literal verb + noun combinations (e.g. *make a cake*) or idioms (e.g. *make a meal*) and verb-particle constructions might follow the same pattern as a verb with a prepositional complement (*take on the task* or *sit on the chair*), which yields that their identification cannot be based on solely syntactic patterns. On the other hand, they are syntactically flexible, that is, they can manifest in various forms: the verb can be inflected, and the noun in light verb constructions can occur in its plural form or can be modified. The verbal component and the noun or the particle may not even be adjacent in e.g. passive sentences or with a pronominal object. However, for higher level applications (such as information extraction or machine translation) it is necessary to treat them as one unit, thus, their automatic identification is desirable.

## 3 Related Work

In the following, methods developed for identifying light verb constructions and verb-particle constructions are summarized shortly.

Cruys and Villada Moirón [5] describe a semantic-based method for identifying verb-preposition-noun combinations in Dutch. Their method relies on selectional preferences for both the noun and the verb and they also make use of automatic noun clustering when considering the selection of semantic classes of nouns for each verb. Cook et al. [4] differentiate between literal and idiomatic usages of verb and noun constructions in English. Their basic hypothesis is that the canonical form of each construction occurs mostly in idioms since they show syntactic variation to a lesser degree than constructions in literal usage. Hence, they make use of syntactic fixedness of idioms when developing their unsupervised method. Bannard [3] seeks to identify verb and noun constructions in English on the basis of syntactic fixedness. He examines whether the noun can have a determiner or not, whether the noun can be modified and whether the construction can have a passive form, which features are exploited in the identification of the constructions. Samardžić and Merlo [12] analyze English and German light verb constructions in parallel corpora: they pay special attention to their manual and automatic alignment. They found that linguistic features (i.e. the degree of compositionality) and the frequency of the construction both have an effect on aligning the constructions.

Baldwin and Villavicencio [2] detect verb-particle constructions in raw texts. They make use of POS-tagging and chunking when developing their classifier while frequency and lexical information are also incorporated in their system. Kim and Baldwin [8] exploit semantic information when deciding whether verb-preposition pairs are verb-particle constructions or not. The (non-)compositionality of verb-particle combinations has been also paid attention in the literature. McCarthy et al. [10] implemented a method to indicate the compositionality of phrasal verbs and Baldwin [1] describes a dataset in which non-compositional VPCs can be found. Methods to extend the coverage of available VPC resources are proposed in [16].

## 4 Experiments

For the automatic identification of verbal collocations, we implemented several rule-based methods, which we describe below in detail.

### 4.1 Background

In order to identify multiword expressions, simple methods are worth examining, which can later serve as a basis for implementing more complex systems. Morphological information can be also exploited in the case of e.g. light verb constructions (the deverbal suffix of the noun may imply that it forms a light verb construction with the verb). Syntactic patterns can be also applied in identifying more complex or syntactically more flexible multiword expressions (e.g. some idioms can be passivized, compare *Who let the cat out of the bag?* and *The cat was let out of the bag*).

Although earlier studies on the detection of verbal collocations generally take syntactic information as a starting point (e.g. [4,3,10,14]), that is, their goal is to classify constructions selected on the basis of syntactic patterns as literal or idiomatic, we would like to identify light verb constructions and verb-particle constructions in running text without assuming that syntactic information is necessarily available. Thus, in our investigations, we will pay distinctive attention to the added value of syntactic features on the system's performance. Given that we are not aware of any other corpora annotated for verb-particle combinations and light verb constructions at the same time, we restrict ourselves to rule-based methods since statistical methods would require a lot more data than available in our annotated database (see 4.3).

### 4.2 Methods for Detecting Verbal Collocations

For identifying verbal collocations, we made use of several methods. In the case of 'POS-rules', each n-gram for which the pre-defined patterns (e.g. `VB.? (NN|NNS)` or `VB.? RP`) could be applied was accepted as LVC or VPC. For POS-tagging, we used the Stanford POS Tagger [15]. Since the methods to follow rely on morphological information (i.e. it is required to know which element is the verb, noun or particle), matching the POS-rules is a prerequisite to apply those methods.

The ‘Suffix’ method exploited the fact that many nominal components in light verb constructions are derived from verbs. Thus, in this case only constructions that matched our POS-rules and contained nouns ending in certain derivational suffixes were allowed.

The ‘Most frequent’ (MF) methods relied on the fact that the most common verbs occur typically in verbal collocations (e.g. *do*, *make*, *take*, *give* etc.) Thus, the 15 most frequent verbs (MFV) typical of light verb constructions and the 10 most frequent verbs typical of verb-particle combinations were collected and constructions that matched our POS-rules and where the stem of the verbal component was among those of the most frequent ones were accepted. The 20 most frequent particles (MFP) were similarly listed and the particle of the VPC candidate had to be among them.

The ‘Stem’ method pays attention to the stem of the noun. In the case of light verb constructions, the nominal component is typically one that is derived from a verbal stem (*make a decision*) or coincides with a verb (*have a walk*). In this case, we accepted only candidates that had a nominal component whose stem was of verbal nature, i.e. coincided with a stem of a verb.

Syntactic information can also be exploited in identifying MWEs. Typically, the syntactic relation between the verb and the nominal component in a light verb construction is `dobj` (using Stanford parser [9]) – if it is a prepositional light verb construction, the relation between the verb and the preposition is `prep`. The relation between a verb and its particle is `prt`. The ‘Syntax’ method accepts candidates among whose members the above syntactic relations hold.

We also combined the above methods to identify noun compounds and light verb constructions in our databases (the union of candidates yielded by the methods is denoted by $\cup$ while the intersection is denoted by $\cap$ in the respective tables). Rule-based methods were evaluated on our Wikipedia database and results are presented in 4.3.

### 4.3 Results

For the evaluation of our models, we developed a corpus of 50 Wikipedia articles, in which several types of multiword expressions (including verb-particle combinations and light verb constructions) and named entities were marked. The database contains 446 occurrences of verb-particle combinations and 368 occurrences of light verb constructions in 4350 sentences and can be downloaded under the Creative Commons license at `http://www.inf.u-szeged.hu/rgai/mwe`.

Results on the rule-based identification of light verb constructions can be seen in Table 1. The recall of the baseline (POS-rules) is high, however, its precision is low (i.e. not all of the candidates defined by the POS patterns are light verb constructions). The ‘Most frequent verb’ (MFV) feature proves to be the most useful: the verbal component of the light verb construction is lexically much more restricted than the noun, which is exploited by this feature. The other two features put some constraints on the nominal component, which is typically of verbal nature in light verb constructions: ‘Suffix’ simply requires the noun to end in a given n-gram (without exploiting further grammatical information) whereas 'Stem' allows nouns derived from verbs. When combining a verbal and a nominal feature, union results in high recall (the combinations typical verb + non-deverbal noun or atypical verb + deverbal noun are also found) while intersection yields high precision (typical verb + deverbal noun combinations are found only).

**Table 1.** Results of rule-based methods for light verb constructions in terms of precision (P), recall (R) and F-measure (F). POS-rules: matching of POS-patterns, Suffix: the noun ends in a given suffix, MFV: the verb is among the 15 most frequent light verbs, Stem: the noun is deverbal.

| Method | P | R | F | F with syntax |
|---|---|---|---|---|
| POS-rules | 7.02 | 76.63 | **12.86** | 16.56 |
| Suffix | 9.62 | 16.3 | 12.1 | 13.11 |
| MFV | 33.83 | 55.16 | **41.94** | **45.31** |
| Stem | 8.56 | 50.54 | 14.64 | 17.96 |
| Suffix ∩ MFV | 44.05 | 10.05 | 16.37 | 15.42 |
| Suffix ∪ MFV | 19.82 | 61.41 | 29.97 | 33.92 |
| Suffix ∩ Stem | 10.35 | 11.14 | 11.1 | 11.68 |
| Suffix ∪ Stem | 8.87 | 57.61 | 15.37 | 18.88 |
| MFV ∩ Stem | 39.53 | 36.96 | 38.2 | 39.81 |
| MFV ∪ Stem | 10.42 | 68.75 | 18.09 | 22.15 |
| Suffix ∩ MFV ∩ Stem | **47.37** | 7.34 | 12.7 | 11.96 |
| Suffix ∪ MFV ∪ Stem | 10.16 | **72.28** | 17.82 | 21.89 |

The added value of syntax was also investigated for LVC detection. As represented in the last column in Table 1, syntax clearly helps in identifying LVCs – except for two cases but its overall effect is to add up to 4% to the F-score. The best result, again, is yielded by the MFV method, which is about 30% above the baseline.

**Table 2.** Results of rule-based methods for verb-particle constructions in terms of precision (P), recall (R) and F-measure (F). POS-rules: matching of POS-patterns, syntax: matching of syntactic patterns, MFV: the verb is among the 10 most frequent verbs, MFP: the particle is among the 20 most frequent particles.

| Method | P | R | F |
|---|---|---|---|
| POS-rules | 29.64 | **64.8** | 40.68 |
| Syntax | 91.89 | 53.36 | 67.52 |
| POS ∩ syntax | 92.12 | 49.78 | 64.63 |
| MFV | 41.96 | 10.54 | 16.85 |
| MFV ∩ syntax | 91.43 | 7.17 | 13.31 |
| MFP | 91.26 | 58.52 | **71.31** |
| MFP ∩ syntax | 93.59 | 49.1 | 64.41 |
| MFV ∪ MFP | 75.07 | **60.09** | 66.75 |
| MFV ∪ MFP ∩ syntax | 92.8 | 49.1 | 64.22 |
| MFV ∩ MFP | **97.56** | 8.97 | 16.43 |
| MFV ∩ MFP ∩ syntax | 96.97 | 7.17 | 13.36 |

When identifying verb-particle constructions simply with POS-patterns, we get a baseline of 40.68 (F-score). Except for the 'Most frequent verb', each method can improve results as represented in Table 2. MFP proves to be the best among the methods, which is due to the high precision of the method. When the two 'Most frequent' methods are contrasted, it is revealed that within a verb-particle construction, the particle seems to be lexically more restricted than the verb, thus, imposing constraints on the former leads to better results while the performance (especially recall) seriously declines when applying the MFV method. On the other hand, the intersection of the two methods yields the highest precision (constructions with a typical verb and a typical particle are only identified) while their union leads to the highest recall (except for the baseline method) since typical verb-atypical particle and atypical verb-typical particle pairs are also found besides typical verb-typical particle pairs.

The analysis of the added value of syntactic features reveals that although syntax proves to be the second best method, when combining it with other features, the overall performance of the system usually declines, however, precision improves a lot. This phenomenon might be connected to potential parsing errors where the parser fails to recognize the `prt` dependency relation between the particle and the verb, thus recall decreases. There is only one exception where the effect of syntax is the opposite: in the case of POS-rules, syntax obviously helps to identify VPCs. This can be expected since due to the common errors in POS-tagging, we chose to include particles and adverbs in our POS-patterns (the difficulties of distinguishing these types of parts of speech are also highlighted in [13]). Whereas this decision results in high recall values, precision seriously degrades, thus, a big pool of VPC candidates is yielded in this way from which the other methods (e.g. syntax) can select true positives.

## 5 Discussion

It is worth contrasting the results achieved for light verb constructions and verb-particle constructions. Making use of only POS-rules does not seem to be satisfactory for LVC detection. However, the most useful feature for identifying LVCs, namely, MFV proves to perform poorly for VPCs, which reflects that the verbal component of LVCs is lexically more restricted than the verbal part of VPCs. However, it is the particle in VPCs that is lexically more restricted as opposed to the verb, which is illustrated by the fact that the method MFP performs best.

As for light verb constructions, the feature 'Stem' seems to be beneficial for recall and this feature can be further enhanced since in some cases, the Porter stemmer did not render the same stem to derivational pairs such as *assumption – assume*, e.g. wordnet-based derivational information might contribute to performance.

Concerning syntactic information, it has clearly positive effects on LVC identification, however, its added value is not unequivocal in the case of verb-particle constructions. Due to possible parsing errors, syntactic features seem to introduce some noise in the performance of the system, thus, the combination of lexical and morphological features (POS-tagging) proves to be the most successful for identifying VPCs because of the relation between the two parts of the construction is rather lexical in nature (not syntactic in the sense that they constitute two separate phrases). On the other hand, light verb constructions do form a syntactic phrase (i.e. their parts can behave as separate phrases) hence syntactic features can be more successfully applied in their identification.

## 6 Conclusions

In this paper, we aimed at identifying verb-particle constructions and light verb constructions in running texts with rule-based methods and compared the effect of several features on performance. For verb-particle constructions, POS-tagging and restriction on the particle seem to yield the best result whereas the combination of POS-tagging, syntactic information and restrictions on the nominal and verbal component have the most beneficial effect on identifying light verb constructions. Special attention was paid to the role of syntax in identifying those types of multiword expressions: although it rather harms performance in the case of verb-particle constructions when combined with other features, it proves effective when applied alone and it is unambiguously helpful for identifying light verb constructions. As future work, we plan to further improve our methods by extending the set and scope of features and refining POS- and syntactic rules. We believe that detecting verb-particle constructions and light verb constructions (i.e. identifying multiword semantic units) can be successfully exploited in several applications in the fields of machine translation or information extraction.

## References

1. Baldwin, T.: A resource for evaluating the deep lexical acquisition of English verb-particle constructions. In: Proceedings of the LREC 2008 Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008), Marrakech, Morocco, pp. 1–2 (2008)
2. Baldwin, T., Villavicencio, A.: Extracting the unextractable: a case study on verb-particles. In: Proceedings of the 6th Conference on Natural Language Learning, pp. 1–7. ACL (2002)
3. Bannard, C.: A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In: Proceedings of the Workshop on a Broader Perspective on Multiword Expressions, pp. 1–8. ACL, Prague (2007)
4. Cook, P., Fazly, A., Stevenson, S.: Pulling their weight: exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In: Proceedings of the Workshop on a Broader Perspective on Multiword Expressions, pp. 41–48. ACL, Prague (2007)
5. Van de Cruys, T., Villada Moirón, B.: Semantics-based multiword expression extraction. In: Proceedings of the Workshop on a Broader Perspective on Multiword Expressions, pp. 25–32. ACL, Prague (2007)
6. Diab, M., Bhutada, P.: Verb Noun Construction MWE Token Classification. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, pp. 17–22. ACL, Singapore (2009)
7. Fazly, A., Stevenson, S.: Distinguishing Subtypes of Multiword Expressions Using Linguistically-Motivated Statistical Measures. In: Proceedings of the Workshop on A Broader Perspective on Multiword Expressions, pp. 9–16. ACL, Prague (2007)
8. Kim, S.N., Baldwin, T.: Automatic identification of English verb particle constructions using linguistic features. In: Proceedings of the Third ACL-SIGSEM Workshop on Prepositions, pp. 65–72 (2006)
9. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: Proceedings of ACL 2003, pp. 423–430. ACL, Sapporo (2003)

10. McCarthy, D., Keller, B., Carroll, J.: Detecting a Continuum of Compositionality in Phrasal Verbs. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, pp. 73–80. ACL, Sapporo (2003)
11. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword Expressions: A Pain in the Neck for NLP. In: Gelbukh, A. (ed.) Proceedings of Conference on Intelligent Text Processing and Computational Linguistics 2002, Mexico City, pp. 1–15 (2002)
12. Samardžić, T., Merlo, P.: Cross-Lingual Variation of Light Verb Constructions: Using Parallel Corpora and Automatic Alignment for Linguistic Research. In: Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground, pp. 52–60. ACL, Uppsala (2010)
13. Santorini, B.: Part-of-speech tagging guidelines for the Penn Treebank Project. Technical report, Department of Computer and Information Science, University of Pennsylvania (1990)
14. Tan, Y.F., Kan, M.-Y., Cui, H.: Extending corpus-based identification of light verb constructions using a supervised learning framework. In: Proceedings of the EACL Workshop on Multi-Word Expressions in a Multilingual Contexts, pp. 49–56. ACL, Trento (2006)
15. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of EMNLP 2000, pp. 63–70. ACL, Hong Kong (2000)
16. Villavicencio, A.: Verb-Particle Constructions and Lexical Resources. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, pp. 57–64. ACL, Sapporo (2003)

# Initialization of fMLLR with Sufficient Statistics from Similar Speakers[*]

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,machlica,muller}@kky.zcu.cz

**Abstract.** One of the most utilized adaptation techniques is the feature Maximum Likelihood Linear Regression (fMLLR). In comparison with other adaptation methods the number of free parameters to be estimated significantly decreases. Thus, the method is well suited for situations with small amount of adaptation data. However, fMLLR still fails in situations with extremely small data sets. Such situations can be solved through proper initialization of fMLLR estimation adding some a-priori information. In this paper a novel approach is proposed solving the problem of fMLLR initialization involving statistics from speakers acoustically close to the speaker to be adapted. Proposed initialization suitably substitutes missing adaptation data with similar data from a training database, fMLLR estimation becomes well-conditioned, and the accuracy of the recognition system increases even in situations with extremely small data sets.

**Keywords:** fMLLR, adaptation, sufficient statistics, speech recognition, robustness, initialization.

## 1 Introduction

Nowadays, in Automatic Speech Recognition (ASR) systems speaker adaptation of an acoustics model represents a standard approach how to improve the accuracy of an ASR system. The most widely used method is feature Maximum Likelihood Linear Regression (fMLLR), which transforms acoustic features for better fit to a Speaker Independent (SI) model.

fMLLR tries to find a linear transformation ($N \times N$ matrix, where $N$ is the features dimension) of an acoustic space, which maximizes probability of test data given a SI model. In the case where small amount of adaptation data is available (especially in on-line recognition) the number of free parameters ($N \times N$) is too high to be properly estimated. Transformation matrix becomes ill-conditioned, and can lead to poor recognition. Some solutions of the problem were already proposed, e.g. lower the number of free parameters using diagonal matrices [1], eigenspace approach [2], or initialization of the estimation [3]. Initialization methods suppress the influence of adaptation data

[*] This research was supported by the Ministry of Education of the Czech Republic project No. MŠMT LC536, by 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.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 187–194, 2011.

for the benefit of initialization data. Usually a compromise has to be made between safety and accuracy of the adaptation. Next problem to solve is how to choose suitable initialization data.

In this paper a similar approach to [4] is used, where prearranged data statistics from similar speakers are utilized to perform an additional EM (Expectation-Maximization) iteration of the SI model according to given statistics, see Section 4. We use the same statistics, but in order to initialize the fMLLR estimation, what proves to be efficient mainly in cases with extremely small amount of adaptation data. The selection of closest (most similar) speakers to the test speaker is crucial and is described in more detail in Section 5. Results of the proposed method compared with standard (basic) fMLLR and unadapted SI system for different sizes of adaptation data can be found in Section 6.

## 2 Adaptation

The adaptation adjusts the SI model so that the probability of the adaptation data would be maximized. The most widely used methods for adaptation are Maximum A-posteriori Probability (MAP) technique and Linear Transformations (LTs). Adaptation techniques do not access the data directly, but only through accumulated statistics, which is the first step preceding the adaptation process. Instead of storing a huge amount of data to estimate the adaptation formulas, adaptation methods need only following statistics:

$$\gamma_{jm}(t) = \frac{\omega_{jm} p(\boldsymbol{o}_t|jm)}{\sum_{m=1}^{M} \omega_{jm} p(\boldsymbol{o}_t|jm)} \tag{1}$$

denoting the $m-th$ mixture's posterior of the $j-th$ state of the HMM,

$$c_{jm} = \sum_{t=1}^{T} \gamma_{jm}(t) \tag{2}$$

representing the soft count of mixture $m$,

$$\boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}) = \sum_{t=1}^{T} \gamma_{jm}(t)\boldsymbol{o}_t \;, \quad \boldsymbol{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) = \sum_{t=1}^{T} \gamma_{jm}(t)\boldsymbol{o}_t\boldsymbol{o}_t^{\mathrm{T}} \tag{3}$$

denoting the sum of the first and the second moment of features aligned to mixture $m$ in the $j$-th state of the HMM.

## 3 Feature Maximum Likelihood Linear Regression (fMLLR)

fMLLR technique belongs to the category of Linear Transformations (LTs), another LT based method is Maximum Likelihood Linear Regression (MLLR). These methods try to find a linear transformation in order to match adaptation data with an acoustics model. Contrary to MAP, LTs can adapt more model components at once using the same transformation (e.g. only one matrix for all the model means), thus they require lower amount of adaptation data since number of free parameters to be estimated is low. Similar model components are clustered into clusters $K_n, n = 1, \ldots, N$ in order to lower the number of adapted parameters [7]. Advantage of fMLLR is that it transforms directly acoustics features instead of an acoustics model (this is the case for MLLR), what is less time-consuming. fMLLR transforms 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 estimation formulas of rows of $\boldsymbol{W}_{(n)}$ are given as

$$\boldsymbol{w}_{(n)i} = \boldsymbol{G}_{(n)i}^{-1}\left(\frac{\boldsymbol{v}_{(n)i}}{\alpha_{(n)}} + \boldsymbol{k}_{(n)i}\right)\,, \tag{6}$$

where $\boldsymbol{v}_{(n)i}$ is the $i$-th row vector of cofactors of matrix $\boldsymbol{A}_{(n)}$, $\alpha_{(n)}$ can be found as a solution of a quadratic function defined in [8],

$$\boldsymbol{k}_{(n)i} = \sum_{m\in K_n} \frac{\mu_{mi}\boldsymbol{\varepsilon}_m(\boldsymbol{\xi})}{\sigma_{mi}^2}\,, \quad \boldsymbol{G}_{(n)i} = \sum_{m\in K_n} \frac{\boldsymbol{\varepsilon}_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}})}{\sigma_{mi}^2}\,, \tag{7}$$

where $\boldsymbol{G}_{(n)i}$, $\boldsymbol{k}_{(n)i}$ are accumulation matrices of statistics (3) of all mixtures $m$ contained in a given cluster $K_n$, and

$$\boldsymbol{\varepsilon}_m(\boldsymbol{\xi}) = \left[\boldsymbol{\varepsilon}_m^{\mathrm{T}}(\boldsymbol{o}), c_m\right]^{\mathrm{T}}\,, \quad \boldsymbol{\varepsilon}_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}}) = \begin{bmatrix} \boldsymbol{\varepsilon}_m(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) & \boldsymbol{\varepsilon}_m(\boldsymbol{o}) \\ \boldsymbol{\varepsilon}_m^{\mathrm{T}}(\boldsymbol{o}) & c_m \end{bmatrix}\,. \tag{8}$$

Equation (6) is a solution of the minimization problem with auxiliary function given in [8]. Matrices $\boldsymbol{A}_{(n)}$ and $\boldsymbol{b}_{(n)}$ are estimated iteratively, thus they have to be suitably initialized (for enough data e.g. randomly, otherwise see Section 5).

In order to accumulate (7) each frame has to be assigned to a HMM state in the first place. In the case of unsupervised fMLLR, the assignment process (more precisely the recognition) has to be done utilizing the not adapted SI system. Since the recognition may contain errors it is suitable to assign a Certainty Factor (CF $\in \langle 0, 1\rangle$) to each of the recognized phones/words/sentences/etc. We work on the word level, CFs for any particular word sequence are extracted from the lattice and may be computed as in [9]. We use only the best path in the lattice. Only the data which transcriptions have high CF (greater than an empirically specified threshold) are used for the adaptation. Still some problems may occur. Even if the CF of a word is high, the boundaries of the word can be inaccurate, because of low values of CF of neighborhood words. Hence, it is useful to take into account the left and right context of each word in the sense of CF. We are seeking for a sequence of three words, where each of them has a CF higher than the threshold and for adaptation we consider only the middle one.

## 4 Sufficient Statistics of Closest Speakers

This method was proposed in [4]. It consists in selecting a subset of speakers who are close in the acoustic space to a given test speaker $t$ whose speech is going to be recognized. Model of the $t$ - th speaker is then estimated according to the already stored HMM data statistics of selected speakers. Compared to the Cluster Adaptive Training (CAT) [5], [6] this method can result in more suitable clusters, because the clusters are determined according to the data of an actual test speaker, hence on-line. This method consists of three steps – accumulation of statistics, cohort selection and estimation of a new model (see Fig.1).

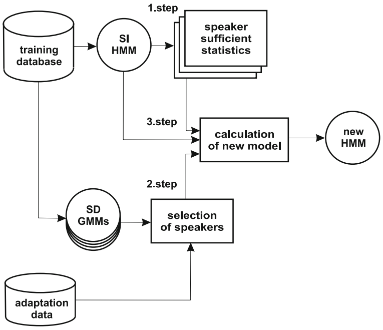


**Fig. 1.** Three steps in the process of a SI model reestimation based on sufficient statistics of closest speakers. SI stands for Speaker Independent model, SD for Speaker Dependent model.

### 4.1 Accumulation of Sufficient Statistics

In the first step, given a SI model sufficient statistics (1)-(3) are accumulated for each speaker in the training database using all his speech data. Hence a set of sufficient statistics is acquired. This step can be done off-line, before the adaptation itself.

### 4.2 Selecting a Cohort of Speakers

In the second step $N$-best speakers are selected from the training database according to their closeness to the test speaker $t$. The selection is performed on-line utilizing $t$ - th speaker's present data. For each of the training speakers a 64 mixture GMM is trained from all their available data. Then, $t$ - th speakers data are scored against each of the GMMs. Verification scores are sorted and $N$-best speakers with highest likelihoods are included into the cohort.

### 4.3 Estimating a New Model

The statistics from the speakers in the cohort are used to reestimate the SI model in order to better match the $t$-th speaker actual data. In [4] one EM iteration of the HMM is performed.

## 5 fMLLR Initialization through Sufficient Statistics

In this section we combine both approaches described above – fMLLR adaptation from Section 3 and statistics of closest speakers from Section 4. We do not train a new model, instead we use the data statistics in order to initialize the fMLLR estimation process. The principle is similar to the steps defined in Section 4 except for a few differences.

### 5.1 Accumulation of Sufficient Statistics

First, for each speaker $s$ in the training set directly matrices $\boldsymbol{k}^s_{(n)i}$ and $\boldsymbol{G}^s_{(n)i}$ given in (7) are accumulated and stored.

### 5.2 Selecting a Cohort of Speakers

Second, we do not compute the likelihood of the whole data set (utterance/recording) of a test speaker $t$ at once. Instead we use a floating window with a fixed size and a fixed shift. All the frames in a window are scored against all the GMMs trained for each of the speakers in the training set and then the window shifts to a new position. For each window position the best scoring GMM is found and the corresponding statistics (matrices $\boldsymbol{k}^s_{(n)i}$ and $\boldsymbol{G}^s_{(n)i}$) are added to the cohort. Thus, the size of the cohort $N_t$ changes for each test speaker $t$, it is not fixed as in [4]. In order to train the GMMs we use MAP adaptation of an Universal Background Model (UBM) [10]. UBM also participates in the scoring of frames in a window, however if UBM scores best nothing is added into the cohort. It should serve to discard non-informative frames.

### 5.3 fMLLR Transform Estimation

Third, fMLLR transformation matrix (5) is computed using all the matrices from the cohort and statistics obtained from the $t$ - th speaker's available data, thus:

$$\boldsymbol{k}_{(n)i} = \textstyle\sum_{s=1}^{N_t} \boldsymbol{k}^s_{(n)i} + \boldsymbol{k}^t_{(n)i} \ , \quad \boldsymbol{G}_{(n)i} = \textstyle\sum_{s=1}^{N_t} \boldsymbol{G}^s_{(n)i} + \boldsymbol{G}^t_{(n)i} \ , \tag{9}$$

for each cluster $n$ and each row $i$ of resulting $\boldsymbol{W}_{(n)}$.

## 6 Experiments

### 6.1 SpeechDat-East (SD-E) Corpus

SpeechDat-East (see [12]) contains telephone speech in 5 languages Czech, Polish, Slovak, Hungarian, and Russian. We used only the Czech part of SD-E. In order to extract the features Mel-frequency cepstral coefficients (MFCC) were utilized, 11 dimensional feature vectors were extracted each 10 ms utilizing a 32 ms hamming window, Cepstral Mean Normalization (CMN) was applied, and $\Delta$, $\Delta^2$ coefficients were added.

A 3 state HMM based on triphones with 2105 states total and 8 GMM mixtures with diagonal covariances in each of the states was trained on 700 speakers with 50 sentences for each speaker (cca 5 sec. for sentence). Using the same data 256 mixture UBM was trained and subsequently all the GMMs of individual speakers were MAP adapted.

To test the systems performance different 200 speakers from SDE were used with 50 sentences for each speaker, however 12 sentences maximal were used for the adaptation. For the recognition a language model based on trigrams was considered [11]. The vocabulary consisted of 7000 words.

### 6.2 Adaptation Setup

In our experiments we utilized unsupervised fMLLR adaptation. At first, the SI model was used to get the word transcription with assigned CFs (as described in Section 3) of given sentences. At the same time a cohort containing statistics was determined as described in Section 5. The window size was set to 30 frames with 10 frames shift resulting in cca 20 speakers per cohort. Note that in a case when several sentences of a test speaker were available we used only the first sentence to determine the cohort. This makes the adaptation process significantly faster preserving sufficient amount of statistics in the cohort (assuming 5 sec. sentences and floating window of size 30 frames with 10 frames shift) – good compromise between good results and small time consumption (important in on-line recognition). At the end, statistics from the cohort along with statistics of given sentences of a test speaker corresponding to words with adequate CFs (see Section 3) were used to estimate the fMLLR transformation matrices. CF threshold was set to 0.99, usable length of one adaptation sentence is then shorter than proclaimed 5 sec. (cca 3 sec.). At the end, the given sentences were once again recognized with adapted model.

Clustering of model components was performed via a regression tree. The threshold for occupation of nodes in the regression tree was set to 1000. In basic (standard) fMLLR the number of transformation matrices depends on the number of adaptation sentences. In the case of fMLLR initialized with sufficient statistics of $N_t$-best speakers ($N_t$best-fMLLR) one can determine the number of matrices in advance since there is always enough adaptation data guaranteed by the cohort. For the $N_t$best-fMLLR 32 transformation matrices for each speaker were created. Only one iteration of fMLLR was carried out.

### 6.3 Results

Results of experiments on basic fMLLR and $N_t$best-fMLLR in dependence on varying number of adaptation sentences can be found in Table 1 and in Figure 2. Results show poor performance of basic fMLLR on small sets of adaptation data, where the estimation of transformation parameters is ill-conditioned (for 5 and less adaptation

**Table 1.** Accuracy (Acc)[%] of the unadapted SI system (baseline), adapted system with basic fMLLR and fMLLR initialized by sufficient statistics from $N_t$best speakers ($N_t$best-fMLLR) in dependence on the number of adaptation sentences. Real system combines basic $N_t$best-fMLLR and fMLLR, $N_t$best-fMLLR is replaced by fMLLR after cca 20-30 sec. of adaptation speech.

| Number of sentences | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|---|---|---|
| unadapted SI system | 62.69 | 62.69 | 62.69 | 62.69 | 62.69 | 62.69 | 62.69 | 62.69 | 62.69 |
| basic fMLLR | 12,31 | 50.49 | 59.73 | 60.62 | 62.03 | 62.82 | 63.71 | 64.19 | 64.25 |
| $N_t$best-fMLLR | 62.77 | 63.37 | 63.31 | 62.96 | 63.13 | 63.22 | 63.23 | 63.54 | 63.65 |
| real system | 62.77 | 63.37 | 63.31 | 62.96 | 63.13 | 63.22 | 63.71 | 64.19 | 64.25 |

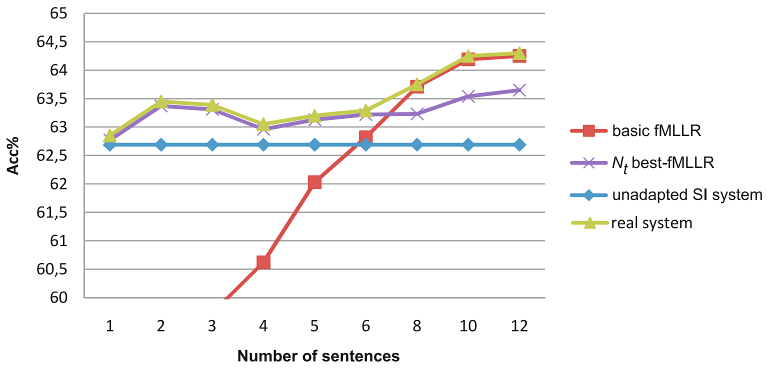


**Fig. 2.** Accuracy (Acc)[%] of systems in dependence on the number of adaptation sentences, see also Table 1

sentences). This is not true for $N_t$best-fMLLR, which actually outperforms the baseline even for small amounts of data. Hence, statistics from closest speakers in the cohort ensure proper initialization and can be thought as a good approximation of statistics of utterances of the test speaker. When the number of sentences (speech data) increases, basic fMLLR becomes better than the $N_t$best-fMLLR.

This is caused by the fact that the overall statistics composed of the test speaker's statistics and statistics of cohort speakers make the final transform more speaker independent than the basic fMLLR. The solution is simple, a threshold has to be determined where the amount of data is sufficient in order to discard the cohort statistics from the estimation process (real system in Figure 2).

## 7 Conclusion

In this paper we proposed an initialization method based on sufficient statistics from $N_t$best speakers ($N_t$best-fMLLR) in order to prevent the poor performance of basic fMLLR in cases of small adaptation data sets. Proposed method uses off-line accumulated statistics form closest speakers in the acoustic space to stabilize the estimation of transformation matrices. Every initialization has to cope with a compromise between better accuracy and well-conditioned adaptation since initialization data hold back the impact of adaptation data. Results show that assuming only few adaptation data $N_t$best-fMLLR outperforms basic fMLLR, because missing data are replaced by $N_t$ best speaker's sufficient statistics. For two adaptation sentences $N_t$best-fMLLR improves the accuracy by 0.6% absolutly compared to unadapted SI system, while fMLLR worstens the performance. Assuming more than 6 adaptation sentences fMLLR gains the lead since the $N_t$ best statistics make the adaptation more speaker independent. Hence, it is useful to lower the number of initialization statistics according to the quantity of adaptation data.

## References

1. Gales, M.J.F.: Maximum likelihood linear transformations for HMM-based speech recognition. Computer Speech and Language 12, 75–98 (1997)
2. Chen, K., Liau, W., Wang, H., Lee, L.: Fast speaker adaptation using eigenspace-based maximum likelihood linear regression. In: International Conference on Spoken Language Processing, Beijing, China, pp. 742–745 (2000)
3. Li, Y., et al.: Incremental on-line feature space MLLR adaptation for telephony speech recognition. In: International Conference on Spoken Language Processing, Denver (2002)
4. Yoshizawa, S., Baba, A., Matsunami, K., Mera, Y., Yamada, M., Shikano, K.: Unsupervised speaker adaptation based on sufficient HMM statistics of selected speakers. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 341–344 (2001)
5. Gales, M.J.F.: Cluster adaptive training of hidden Markov models. IEEE Transactions on Speech and Audio Processing, 417–428 (2000)
6. Vaněk, J., Psutka, J., Zelinka, J., Trmal, J.: Training of speaker-clustered acoustic models for use in real-time recognizers. In: Sigmap 2009, Milan, pp. 131–135 (2009)
7. Gales, M.J.F.: The generation and use of regression class trees for MLLR adaptation. Cambridge University Engineering Department, Cambridge (1996)
8. Povey, D., Saon, G.: Feature and model space speaker adaptation with full covariance Gaussians, Interspeech, paper 2050-Tue2BuP.14 (2006)
9. Uebel, L.F., Woodland, P.C.: Improvements in linear transform based speaker adaptation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 49–52 (2001)
10. Reynolds, D. A., Quatieri, T. F., Dunn, R. D.:Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing, 19–41 (2000)
11. Pražák, A., Psutka, J., Hoidekr, J., et al.: 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)
12. Pollak, P., et al.: SpeechDat(E) - Eastern European Telephone Speech Databases. In: XLDB - Very Large Telephone Speech Databases (ELRA), Paris (2000)

# Intelligibility Rating with Automatic Speech Recognition, Prosodic, and Cepstral Evaluation

Tino Haderlein[1,2], Cornelia Moers[3],
Bernd Möbius[4], Frank Rosanowski[2], and Elmar Nöth[1]

[1] University of Erlangen-Nuremberg, Pattern Recognition Lab (Informatik 5),
Martensstraße 3, 91058 Erlangen, Germany
Tino.Haderlein@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de
[2] University of Erlangen-Nuremberg, Department of Phoniatrics and Pedaudiology,
Bohlenplatz 21, 91054 Erlangen, Germany
[3] University of Bonn, Department of Speech and Communication,
Poppelsdorfer Allee 47, 53115 Bonn, Germany
[4] Saarland University, Department of Computational Linguistics and Phonetics,
Postfach 151150, 66041 Saarbrücken, Germany

**Abstract.** For voice rehabilitation, speech intelligibility is an important criterion. Automatic evaluation of intelligibility has been shown to be successful for automatic speech recognition methods combined with prosodic analysis. In this paper, this method is extended by using measures based on the Cepstral Peak Prominence (CPP). 73 hoarse patients ($48.3 \pm 16.8$ years) uttered the vowel /e/ and read the German version of the text “The North Wind and the Sun”. Their intelligibility was evaluated perceptually by 5 speech therapists and physicians according to a 5-point scale. Support Vector Regression (SVR) revealed a feature set with a human-machine correlation of $r = 0.85$ consisting of the word accuracy, smoothed CPP computed from a speech section, and three prosodic features (normalized energy of word-pause-word intervals, $F_0$ value at voice offset in a word, and standard deviation of jitter). The average human-human correlation was $r = 0.82$. Hence, the automatic method can be a meaningful objective support for perceptual analysis.

## 1 Introduction

Chronic voice diseases cause enormous costs for modern communication society [14]. A standardized, efficient method for voice assessment is therefore needed. Despite many attempts for automation, perception-based methods are still the basis for the evaluation of voice pathologies. This, however, is too inconsistent among single raters to establish a standardized and unified classification.

Perception experiments are usually applied to spontaneous speech, standard sentences, or standard texts. Automatic analysis relies mostly on sustained vowels [11]. The advantage of speech recordings is that they contain phonation onsets, variation of $F_0$ and pauses [13]. Furthermore, they allow to evaluate speech-related criteria, such as intelligibility. This paper focuses on automatic intelligibility assessment of chronically hoarse persons by means of automatic speech recognition, prosodic and cepstral analysis.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 195–202, 2011.

Most studies on automatic voice evaluation use perturbation-based parameters, such as jitter, shimmer, or the noise-to-harmonicity ratio (NHR, [11]). However, perturbation parameters have a substantial disadvantage. They require exact determination of the cycles of the fundamental frequency $F_0$. In severe dysphonia it is difficult to find an $F_0$ due to the irregularity of phonation. This drawback can be eliminated by using the Cepstral Peak Prominence (CPP) and the Smoothed Cepstral Peak Prominence (CPPS) which represent spectral noise. They do not require $F_0$ detection and showed high human-machine correlations in previous studies [1,5,8]. It is obvious that CPP expresses voice quality rather than intelligibility, but these two perceptual criteria are highly correlated with each other in voice pathologies [4]. Hence, CPP may also provide a better modeling of the perceptual concept of intelligibility.

The questions addressed in this paper are the following: How can cepstral-based evaluation support the established evaluation of intelligibility by a speech recognizer and prosodic analysis [4,10]? Are there significant differences between the results of automatic vowel and text evaluation?

In Sect. 2, the audio data and perceptive evaluation will be introduced. Section 3 will give some information about the cepstral analysis, Sect. 4 describes the speech recognizer. An overview of the prosodic analysis and Support Vector Regression will be presented in Sect. 5 and 6, and Sect. 7 will discuss the results.

## 2 Test Data and Subjective Evaluation

73 German persons with chronic hoarseness (24 men and 49 women) between 19 and 85 years of age participated in this study. The average age was 48.3 years with a standard deviation of 16.8 years. Patients suffering from cancer were excluded. Each person uttered the vowel /e/ and read the text “Der Nordwind und die Sonne” (“The North Wind and the Sun”, [9]), a phonetically balanced standard text which is frequently used in medical speech evaluation in German-speaking countries. It contains 108 words (71 distinct) with 172 syllables. The data were recorded with a sampling frequency of 16 kHz and 16 bit amplitude resolution by a microphone AKG C 420.

Five experienced phoniatricians and speech scientists evaluated each speaker’s intelligibility in each recording according to a 5-point scale with the labels “very high”, “high”, “moderate”, “low”, and “none”. Each rater’s decision for each patient was converted to an integer number between 1 and 5. The average of all raters served as the reference for the automatic evaluation.

## 3 Cepstral Analysis

The Cepstral Peak Prominence (CPP) is the logarithmic ratio between the cepstral peak and the regression line over the entire cepstrum at this quefrency (Fig. 1). A strongly distorted voice has a flat cepstrum and a low CPP due to its inharmonic structure. The computation of CPP and the Smoothed Cepstral Peak Prominence (CPPS) was performed by the free software “cpps” [7] which implements the algorithm introduced by Hillenbrand and Houde [8]. The cepstrum was computed for each 10 ms frame, CPPS was averaged over 10 frames and 10 cepstrum bins. The vowel-based results will be denoted by “CPP-v” and “CPPS-v”. For the automatic speech evaluations (“CPP-NW” and “CPPS-NW”), the first sentence only (approx. 8-12 seconds, 27 words, 44 syllables) of the read-out text was used. Sections in which the patients laughed or cleared their throat were removed from the recording.

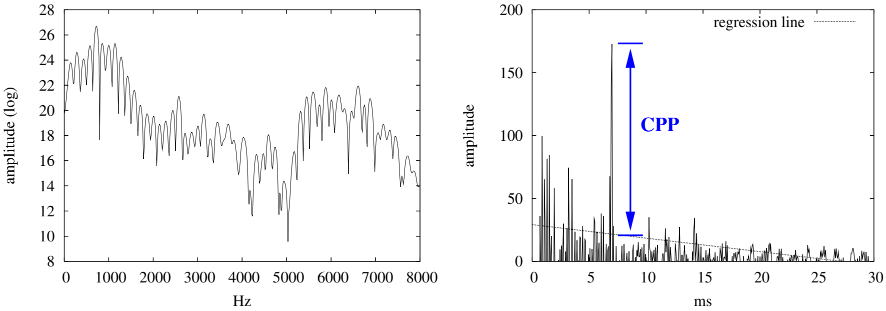


**Fig. 1.** Logarithmic power spectrum *(left)* and cepstrum *(right)* of a vowel section with Cepstral Peak Prominence (CPP)

## 4 The Speech Recognition System

The speech recognition system used for the experiments is described in detail in [17]. It is based on semi-continuous Hidden Markov Models (HMM) and can handle spontaneous speech with mid-sized vocabularies up to 10,000 words. It can model phones in any context size that is 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; the codebook had 500 Gaussians with full covariance matrices. The short-time analysis applies a Hamming window with a length of 16 ms, the frame rate is 10 ms. The filterbank for the Mel-spectrum consists of 25 triangle filters. For each frame, a 24-dimensional feature vector is computed. It contains short-time energy, 11 Mel-frequency cepstral coefficients, and the first-order derivatives of these 12 static features. The derivatives are approximated by the slope of a linear regression line over 5 consecutive frames (56 ms).

The baseline system for the experiments in this paper was trained on German dialogues of non-pathologic speakers from the VERBMOBIL project [18]. The data had been recorded with a close-talking microphone at a sampling frequency of 16 kHz and 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 samples (1042 words) for the validation set, i.e. the corpus partitions were the same as in [17].

The recognition vocabulary of the recognizer was changed to the 71 words of the standard text. The word accuracy and the word correctness were used as basic automatic measures for intelligibility since they had been successful for other voice and speech pathologies [4,10]. They are computed from the comparison between the recognized word sequence and the reference text consisting of the $n_{\mathrm{all}} = 108$ words of the read text. With the number of words that were wrongly substituted ($n_{\mathrm{sub}}$), deleted ($n_{\mathrm{del}}$) and inserted ($n_{\mathrm{ins}}$) by the recognizer, the word accuracy in percent is given as

$$\mathrm{WA} = [1 - (n_{\mathrm{sub}} + n_{\mathrm{del}} + n_{\mathrm{ins}})/n_{\mathrm{all}}] \cdot 100$$

while the word correctness omits the wrongly inserted words:

$$\mathrm{WR} = [1 - (n_{\mathrm{sub}} + n_{\mathrm{del}})/n_{\mathrm{all}}] \cdot 100$$

Only a unigram language model was used so that the results mainly depend on the acoustic models. A higher-order model would correct too many recognition errors and thus make WA and WR useless as measures for intelligibility.

## 5 Prosodic Features

In order to find automatically computable counterparts for intelligibility, also a "prosody module" was used to compute features based upon frequency, duration and speech energy (intensity) measures. This is state-of-the-art in automatic speech analysis on normal voices [3,12,15].

The prosody module processes the output of the word recognition module and the speech signal itself. Hence, the time-alignment of the recognizer and the information about the underlying phoneme classes can be used by the module. For each speech unit of interest (here: words), a fixed reference point has to be chosen for the computation of the prosodic features. This point was chosen at the end of a word because the word is a well–defined unit in word recognition, it can be provided by any standard word recognizer, and because this point can be more easily defined than, for example, the middle of the syllable nucleus in word accent position. For each reference point, 95 prosodic features are computed from 28 base features over intervals which contain one single word, a word-pause-word interval or the pause between two words. A full description of the features used is beyond the scope of this paper; details and further references are given in [2].

In addition to the 95 local features per word, 15 global features were computed from jitter, shimmer and the number of voiced/unvoiced decisions for each 15-word interval. They cover the means and standard deviations for jitter and shimmer, the number, length and maximum length each for voiced and unvoiced sections, the ratio of the numbers of voiced and unvoiced sections, the ratio of length of voiced sections to the length of the signal and the same for unvoiced sections. The last global feature is the standard deviation of the $F_0$.

## 6 Support Vector Regression (SVR)

In order to find the best subset of word accuracy, word correctness, the prosodic features and cepstral measures to model the subjective ratings, Support Vector Regression (SVR, [16]) was used. The general idea of regression is to use the vectors of a training set to approximate a function which tries to predict the target value of a given vector of the test set. Here, the training set are the automatically computed measures, and the test set consists of the subjective intelligibility scores. For this study, the sequential minimal optimization algorithm (SMO, [16]) of the Weka toolbox [19] was applied in a 10-fold cross-validation manner.

The pre-defined correlation-based feature selection algorithm [6] had been altered so that the number of matrix inversions was substantially reduced at the cost of a slightly worse selection result [10, pp. 59-61]. The features with the highest ranks were used as input for the SVR.

## 7 Results and Discussion

The correlations between the perceptual evaluation and single automatic measures are given in Table 1. The human-machine correlations of these measures alone are not as good as the inter-rater correlation of a panel of experts (Table 2). But it appears that WA outperforms WR, and the text-based cepstral measures are clearly better than the vowel-based ones. The correlations are negative because high recognition rates and cepstral peaks came from "good" voices with a low score and vice versa. The values did not change significantly throughout the study when Spearman's rank-order correlation $\rho$ was computed. For this reason, only Pearson's $r$ is given.

By using WA, WR, the CPP measures, and the prosodic features as input for SVR, higher correlations to the subjective intelligibility score were obtained (Table 3). The WR and the vowel-based CPP measures did not appear in the selected feature list. A human-machine correlation of $r = 0.85$ was achieved for the set of WA, CPPS-NW, the normalized energy of word-pause-word intervals (EnNormWPW), the $F_0$ value at the voice offset in a word (F0OffWord), and the standard deviation of jitter (StandDevJitter). With the latter three prosodic features alone, $r = 0.79$ was measured. CPPS-NW and WA together reach $r = 0.83$. The other selected experiments given in Table 3 show that for a human-machine correlation of $r \geq 0.80$ either WA or CPPS-NW are needed in any case.

The energy value EnNormWPW is normalized with respect to a pre-computed speaker list. If the person has a hoarse and irregular voice, then the energy level especially in the high frequency portions is raised. For this reason, this feature may contribute strongly to the best feature set. The impact of the $F_0$ value can be explained by the noisy speech that causes octave errors during $F_0$ detection, i.e. instead of the real fundamental frequency, one of its harmonics is found. With more "noisy speech", this may influence the $F_0$ trajectory and hence the correlation to the subjective results. It is not clear so far, however, why only the end of the voiced sections causes a noticeable effect. There may be a connection to changes in the airstream between the beginning and the end of words or phrases. It may have its reason in the high speaking effort which leads to more irregularities especially in these positions, but this has to be confirmed by more detailed experiments. Jitter is one of the established measures for voice pathology. However, a certain amount of jitter and regular changes thereof are present in normal voices. When changes of jitter over time become irregular, this may also be an indicator for a less intelligible voice. Note that the prosody module computes the $F_0$ and jitter values only on sections which it has previously identified as voiced.

**Table 1.** Subjective and objective evaluation results for 73 hoarse speakers: intelligibility, word accuracy (WA) and word correctness (WR), and the cepstral peak measures; the rightmost column shows the correlation $r$ between the human evaluation and the respective automatic measure

| measure | unit | mean | st. dev. | min. | max. | $r$ |
|---|---|---|---|---|---|---|
| intell. | points | 2.5 | 1.0 | 1.0 | 5.0 | *1.00* |
| WA | % | 69.3 | 14.3 | 27.8 | 90.1 | –0.74 |
| WR | % | 73.5 | 12.0 | 28.9 | 90.1 | –0.69 |
| CPP-v | dB | 17.2 | 4.3 | 8.8 | 25.3 | –0.61 |
| CPPS-v | dB | 6.1 | 2.2 | 0.9 | 11.1 | –0.58 |
| CPP-NW | dB | 12.1 | 1.6 | 9.1 | 16.3 | –0.69 |
| CPPS-NW | dB | 4.1 | 1.0 | 1.9 | 6.3 | –0.74 |

**Table 2.** Inter-rater correlation $r$ for intelligibility between each rater and the average of the remaining raters

| rater | K | R | S | T | V | avg. |
|---|---|---|---|---|---|---|
| $r$ | 0.78 | 0.84 | 0.88 | 0.75 | 0.84 | 0.82 |

**Table 3.** SVR regression weights for the best subset (experiment 1) and selected other subsets, and their correlation $r$ to the subjective intelligibility scores (last row)

| feature | exp. 1 | exp. 2 | exp. 3 | exp. 4 | exp. 5 | exp. 6 | exp. 7 |
|---|---|---|---|---|---|---|---|
| EnNormWPW | 0.228 | 0.980 | 0.840 |  | 0.345 | 0.660 |  |
| F0OffWord | –0.146 | –0.428 |  |  |  |  |  |
| StandDevJitter | 0.167 | 0.522 | 0.549 | 0.168 | 0.343 | 0.178 |  |
| CPPS-NW | –0.412 |  |  | –0.485 |  | –0.524 | –0.632 |
| WA | –0.431 |  |  | –0.579 | –0.532 |  | –0.539 |
| correlation $r$ | 0.85 | 0.79 | 0.74 | 0.84 | 0.83 | 0.81 | 0.83 |

**Table 4.** Correlation of the feature values of the best feature set for all 73 speakers

| feature | F0OffWord | StandDevJitter | CPPS-NW | WA |
|---|---|---|---|---|
| EnNormWPW | –0.03 | 0.23 | –0.56 | –0.74 |
| F0OffWord |  | –0.10 | 0.26 | 0.09 |
| StandDevJitter |  |  | –0.58 | –0.30 |
| CPPS-NW |  |  |  | 0.56 |

The correlations between the feature values of the best subset are given in Table 4. A high EnNormWPW correlates significantly with a low CPPS-NW and a low WA. Likewise, jitter and its standard deviation are higher which correlates negatively with CPPS-NW. The low CPPS-NW in a distorted voice correlates with a low recognition rate.

For this study, patients read a standard text, and voice professionals evaluated intelligibility. It is often argued that intelligibility should be evaluated by an "inverse intelligibility test": The patient utters a subset of words and sentences from a carefully built corpus. A naïve listener writes down what he or she heard. The percentage of correctly understood words is a measure for the intelligibility of the patient. However, when automatic speech evaluation is performed for instance with respect to prosodic phenomena, such as word durations or percentage of voiced segments, then comparable results for all patients can only be achieved when all the patients read the same defined words or text. This means that an inverse intelligibility test can no longer be performed, and intelligibility has to be rated on a grading scale instead.

The results obtained in this study allow for the following conclusions: There is a significant correlation between subjective rating of intelligibility and automatic evaluation. The human-machine correlation is better than the average inter-rater correlation among speech experts. Cepstral-based measures improve the human-machine correlation, but only when they are computed from a speech recording and not from a sustained vowel only. The method can serve as the basis for an automatic, objective system that can support voice rehabilitation.

**Acknowledgments.** This work was partially funded by the German Cancer Aid (Deutsche Krebshilfe) under grant 107873. The responsibility for the contents of this study lies with the authors. We would like to thank Dr. Hikmet Toy for acquiring and documenting the audio data.

## References

1. Awan, S., Roy, N.: Outcomes Measurement in Voice Disorders: Application of an Acoustic Index of Dysphonia Severity. J. Speech Lang. Hear. Res. 52, 482–499 (2009)
2. Batliner, A., Buckow, J., Niemann, H., Nöth, E., Warnke, V.: The Prosody Module. In: Wahlster [18], pp. 106–121
3. Chen, K., Hasegawa-Johnson, M., Cohen, A., Borys, S., Kim, S.S., Cole, J., Choi, J.Y.: Prosody dependent speech recognition on radio news corpus of American English. IEEE Trans. Audio, Speech, and Language Processing 14, 232–245 (2006)
4. Haderlein, T.: Automatic Evaluation of Tracheoesophageal Substitute Voices, Studien zur Mustererkennung, vol. 25. Logos, Berlin (2007)
5. Halberstam, B.: Acoustic and Perceptual Parameters Relating to Connected Speech Are More Reliable Measures of Hoarseness than Parameters Relating to Sustained Vowels. ORL J. Otorhinolaryngol. Relat. Spec. 66, 70–73 (2004)
6. Hall, M.: Correlation-based Feature Subset Selection for Machine Learning. Ph.D. thesis, University of Waikato, Hamilton, New Zealand (1999)
7. Hillenbrand, J.: cpps.exe (software), http://homepages.wmich.edu/~hillenbr (accessed May 30, 2011)
8. Hillenbrand, J., Houde, R.: Acoustic Correlates of Breathy Vocal Quality: Dysphonic Voices and Continuous Speech. J. Speech Hear. Res. 39, 311–321 (1996)
9. International Phonetic Association (IPA): Handbook of the International Phonetic Association. Cambridge University Press, Cambridge (1999)
10. Maier, A.: Speech of Children with Cleft Lip and Palate: Automatic Assessment, Studien zur Mustererkennung, vol. 29. Logos, Berlin (2009)

11. Maryn, Y., Roy, N., De Bodt, M., Van Cauwenberge, P., Corthals, P.: Acoustic measurement of overall voice quality: A meta-analysis. J. Acoust. Soc. Am. 126, 2619–2634 (2009)
12. Nöth, E., Batliner, A., Kießling, A., Kompe, R., Niemann, H.: VERBMOBIL: The Use of Prosody in the Linguistic Components of a Speech Understanding System. IEEE Trans. on Speech and Audio Processing 8, 519–532 (2000)
13. Parsa, V., Jamieson, D.: Acoustic discrimination of pathological voice: sustained vowels versus continuous speech. J. Speech Lang. Hear. Res. 44, 327–339 (2001)
14. Ruben, R.: Redefining the survival of the fittest: communication disorders in the 21st century. Laryngoscope 110, 241–245 (2000)
15. Shriberg, E., Stolcke, A.: Direct Modeling of Prosody: An Overview of Applications in Automatic Speech Processing. In: Proc. International Conference on Speech Prosody, Nara, Japan, pp. 575–582 (2004)
16. Smola, A., Schölkopf, B.: A Tutorial on Support Vector Regression. Statistics and Computing 14, 199–222 (2004)
17. Stemmer, G.: Modeling Variability in Speech Recognition, Studien zur Mustererkennung, vol. 19. Logos, Berlin (2005)
18. Wahlster, W. (ed.): Verbmobil: Foundations of Speech-to-Speech Translation. Springer, Berlin (2000)
19. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

# Maximum Entropy Named Entity Recognition for Czech Language

Michal Konkol and Miloslav Konopík

University of West Bohemia
Laboratory of Intelligent Communication Systems
Univerzitni 8, 30614 Pilsen, Czech Republic
{konkol,konopik}@kiv.zcu.cz

**Abstract.** Named Entity Recognition (NER) is an important preprocessing tool for many Natural Language Processing tasks like Information Retrieval, Question Answering or Machine Translation. This paper is focused on NER for Czech language. The proposed NER is based on knowledge and experiences acquired on other languages and adapted for Czech. Our recognizer outperforms the previously introduced recognizers for Czech. The article is also focused on the use of semantic spaces for NER. Although no significant improvement was yet achieved in this way, we believe that the research is worth of sharing.

**Keywords:** Named Entity Recognition, Maximum Entropy, Semantic Spaces, Czech.

## 1 Introduction

Named Entity recognition (NER) is a very important preprocessing tool for Question Answering, Information Retrieval or Machine Translation. NER was firstly introduced at Message Understanding Conference (MUC) 6 [4] in 1995. The definition of the NER task at MUC-6 was to find 7 categories of Named Entities (NE) like persons, organizations or dates. These expressions are very important, because they are very often the key points in a text. Properly identified NEs can improve results for other Natural Language Processing tasks.

Semantic spaces are one of recent fields of research that focuses on methods to automatically find relations between words. It started with the well known LSA method [2] and continued to the very advanced Beagle method [6]. The idea is to use relations between words to help dealing with an unknown context using a similar known context. The similar known context is found using semantic spaces (see section 5). Many application such as Information Retrieval have already proven the usefulness of semantic spaces. Our intention in this article is to explore the possibility to use semantic spaces for NER.

In this paper a new recognizer based on Maximum Entropy and semantic spaces is presented. The classifier is described in section 3. Details about used features can be found in section 4. Basic information about semantic spaces is given in section 5. Experiments and results are presented in section 6. The last section contains the summary.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 203–210, 2011.

## 2 State of the Art

Since the time of MUC-6 a big effort has been put into NER. Many systems have been presented for different languages. Although the majority of systems was done for English, good systems were introduced for some other languages such as Chinese, Japanese or German [14]. The state of the art F-measure for English is around 90 and for German around 70. This difference around 20 percent is caused by the differences of these languages.

There are two basic approaches to NER. The first one is based on hand made rules and dictionaries and typically involves techniques like Context Free Grammars, Finite State Automata or Regular Expressions. The second one is based on statistical methods. The most used methods for statistical NER are Maximum Entropy Models [1], Conditional Random Fields [11], Hidden Markov Models [16], Support Vector Machines [5] among others. There are also hybrid systems combining more of mentioned methods [7].

There are also different types of training. The training methods can be divided into supervised, semi-supervised and unsupervised methods. Supervised methods need labelled training data to find the best parameterization of the classifier. Semi-supervised methods need some small data which are used as a seed for the training. Unsupervised methods do not need any data.

The Czech language is quite different in comparison with all mentioned languages. Czech is highly inflectional and has a more flexible word order. In addition the custom for writing proper nouns is not very helpful and in some cases not even stable. In comparison to English, the Czech NER is still unexplored. Only two systems for NER were presented for Czech language. The first system used Decision Trees and its F-measure was 68 [15]. The second one was based on SVM and achieved F-measure 71 on the same corpus [8] and was introduced in 2009.

## 3 Classifier

Our classifier is based on the maximum entropy principle. The principle says that we are looking for a model which will satisfy all our constraints and does not make any other assumptions. To define a constraint we firstly need to define a feature. A feature is a function in the following form. Typically binary features are used, but in general any non-negative function is possible. For example, consider the following feature designed to express the relation between NE class PERSON and the capitalization of the word $x$.

$$f(x,y) = \begin{cases} 1 & \text{if } y \text{ is PERSON and } x \text{ starts with capital letter} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The constraint is then defined as equality of mean values for a given feature.

$$E_p(f_i(x,y)) = E_{\tilde{p}}(f_i(x,y)) \tag{2}$$

where $E_{\tilde{p}}(f_i(x,y))$ is mean value of a feature computed over the training data and $E_p(f_i(x,y))$ is mean value of the model. It is guaranteed that such a model exists. In addition it is unique, follows the maximum-likelihood distribution and is in the following form[3].

$$p(y|x) = \frac{1}{Z(x)} \exp \sum_i \lambda_i f_i(x, y) \tag{3}$$

$$Z(x) = \exp \sum_i \lambda_i f_i(x, y) \tag{4}$$

$Z(x)$ is just a normalizing factor and ensures that $p(y|x)$ is a probability distribution. The $\lambda_1 \dots \lambda_n$ are parameters of the model. Various training algorithms can be used for finding appropriate parameters. The most common algorithms are Generalized Iterative Scaling (GIS) and Improved Iterative Scaling (IIS). These algorithms are not the most effective though [10]. We have implemented GIS and a limited memory BFGS (L-BFGS) method [12]. We have to approve that L-BFGS is much more effective than GIS. The classifier for one of the test mentioned later was trained using both methods. The training with L-BFGS method takes 271.053 seconds, while with GIS it was 486.258 seconds. Different stopping conditions were used and L-BFGS should be more precise. The difference in time would be probably higher with the same stopping conditions.

## 4 Features

The recognizer is built from two important parts. The first part is a classifier which was described in the previous section. The second part is a feature set. Both part are equally important. Our feature set is composed of the following features.

- *Morphological features* are used separately or in combinations. One of possible combinations can be if this word has the same case, number and gender as the previous word. We use part of speech, case, person, number and gender as features.
- *Lemma* is almost a must for highly inflectional language like Czech. Many words can have more than 10 different word forms and it causes a sparseness problem without lemmatization. Lemmas are also useful for other features like gazetteers, because they are usually using only lemmas. The F-measure decreases rapidly without usage of lemmas.
- *Regular expressions* are used for some special types of words. Most of the regular expressions are used for identification of numbers, dates and times. Some are used for special words containing mixed capitalization with numbers or symbols or for identification of short cuts.
- *Capitalization* of words has different importance among languages. Capitalization is very important for English but for German is much less important. The Czech language is somewhere between English and German. Three classes of capitalization are used as a feature. Mixed capitalization, all capitalized and first letter capitalized.
- *Gazetteers* are used for some classes. All of our gazetteers are from publicly accessible sources and are not manually edited. We use gazetteers for forenames, surnames, cities, rivers, mountains, organizations etc.

- *Learned lists* are similar to gazetteers but are learned directly from the corpus. This feature is real valued and returns the probability that given word belongs to a particular class.

All the mentioned features are computed for some window around the classified word. A couple of different window sizes were tested. The final system uses a window $-3, \dots, 3$ with an exception of lemmas which have window $-2, \dots, 2$. The difference between $-3, \dots, 3$ and $-2, \dots, 2$ is very small in the final results. Larger windows did not improve results and smaller windows did not achieve similar results.

## 5 Semantic Spaces

Semantic space is a space of words. Every word is a single point in this space and is represented by its vector, which is often quiet large. The position of the word is somehow related to its meaning. After the creation of semantic space a distance between words can be measured. This distance can be used as a relation or similarity of the words.

There are various methods which can build a semantic space from unlabelled data. The methods use different ways to create a semantic space and therefore their results are slightly different. Some of these methods are LSA [2], HAL [9],COALS [13] or BEAGLE [6].

In our experiments we have used the COALS method. This method constructs a word matrix where each position is count of occurrences of particular words. The matrix is then normalized and a Singular Value Decomposition (SVD) is calculated. SVD reduces the dimension of vector space. We have used the dimension reduced to 1000.

Our basic idea is that if some word is similar to a known NE, there is a higher probability that this word is a NE of the same class. This idea can be extended to the context. If we know that a word is almost always person NE if the previous word is ”Mr.”, then it would be probably person NE if there is a word with high similarity to ”Mr.”.

We have tested the following usages of semantic spaces.

- *Bigger groups* – we have created lists of about 20 most frequent words for each NE category. The feature is then the average similarity of the classified word to the words on the list.

$$f(y|x) = \frac{\sum_{w \in G_y} similarity(x, w)}{|G_y|} \qquad \forall y \tag{5}$$

- *Smaller groups* – we have created about 20 smaller groups with 2-4 words. One group was for example Prague, Pilsen and Brno, which are largest Czech cities. The feature is again average similarity to these words.

$$f(y|x) = \frac{\sum_{w \in G_i} similarity(x, w)}{|G_i|} \qquad \forall i \tag{6}$$

- *Single words* – the last feature uses similarity to individual words. Each feature is a similarity to a particular word. This feature is very computationally demanding.

$$f(y|x) = similarity(x, w) \qquad \forall w \in V \tag{7}$$

# 6 Experiments

All experiments were done on the Czech Named Entity corpus [15]. The corpus was divided into two parts. The first part contains 90% of sentences and is used for training. The second part was used as test data. Each experiment was done 10 times with different parts of corpus used as test data. The presented results are averaged.

Our experiments are evaluated by the standard measures for NER which are precision, recall and F-measure (or F-score). These quantities have the following meaning.

- *Precision* is percent of correctly marked entities from all marked entities.
- *Recall* is percent of correctly marked entities from all entities in the data.
- *F-measure* is a harmonic mean of precision and recall.

## 6.1 Results

Previous results for Czech NER are shown in table 1. Our primary goal was to create an effective recognizer and to improve these results.

**Table 1.** A comparison of previous results with our experiments

| | | precision | recall | F-measure |
|---|---|---|---|---|
| Related work | DT [15] | 81 | 59 | 68 |
| | SVM [8] | 75 | 67 | 71 |
| Our experiments | Basic features | 76,78 | 69,58 | 72,94 |
| | Big groups | 76,89 | 69,71 | 73,08 |
| | Small groups | 76,84 | 69,18 | 72,76 |
| | Single words | 76,61 | 69,30 | 72,72 |

The first experiment was made with *basic features* listed in section 4 except semantic spaces. Its purpose was to create a good starting point for other experiments and a preprocessing tool for other projects. This experiment was successful and improved the state of the art for Czech NER. The detailed results are shown in table 2.

The following experiments were focused on semantic spaces. We assumed that entity should be found in similar contexts like other entities in the same class. So we have made a list of 20 typical words for each class (denoted as *bigger groups* in table 2) and made an average similarity of these words with the classified word. The results in table 3 showed that this feature does not work. The problem was identified as too many words in the list of typical words.

We have tried to fix the problem of the second experiment with *smaller groups*. New groups had about 3 words. For example one group was made for Czech cities and contains Prague, Brno, Pilsen and Ostrava, which are four largest cities in Czech republic. There were about 20 groups focused specifically on some subclass. There was no significant change in the results (table 4).

The last experiment leaved the idea of groups and used *single words*. Similarity of each word from the corpus is taken as a feature. This feature is very computational

**Table 2.** Results for basic features

| | precision | recall | F-measure |
|---|---|---|---|
| NUM | 60,54 | 62,02 | 55,98 |
| LOC | 81,08 | 74,97 | 77,82 |
| ORG | 63,33 | 51,62 | 56,76 |
| OTH | 66,29 | 42,40 | 51,48 |
| PER | 81,23 | 86,54 | 83,76 |
| DAT | 88,51 | 87,23 | 87,77 |
| Overall | 76,78 | 69,58 | 72,94 |

**Table 3.** Results for groups of 20 words

| | precision | recall | F-measure |
|---|---|---|---|
| NUM | 60,75 | 62,18 | 56,28 |
| LOC | 80,54 | 75,23 | 77,71 |
| ORG | 63,70 | 52,07 | 57,08 |
| OTH | 67,21 | 41,91 | 51,36 |
| PER | 81,29 | 86,80 | 83,90 |
| DAT | 87,70 | 87,33 | 87,41 |
| Overall | 76,89 | 69,71 | 73,08 |

**Table 4.** Results for groups of 3 words

| | precision | recall | F-measure |
|---|---|---|---|
| NUM | 60,25 | 61,77 | 55,55 |
| LOC | 80,89 | 74,60 | 77,55 |
| ORG | 62,53 | 50,81 | 56,03 |
| OTH | 67,98 | 41,58 | 51,24 |
| PER | 81,42 | 86,48 | 83,83 |
| DAT | 87,69 | 87,50 | 87,46 |
| Overall | 76,84 | 69,18 | 72,76 |

**Table 5.** Results for single words

| | precision | recall | F-measure |
|---|---|---|---|
| NUM | 60,45 | 62,28 | 56,09 |
| LOC | 81,08 | 74,45 | 77,53 |
| ORG | 63,14 | 51,07 | 56,30 |
| OTH | 65,76 | 41,75 | 50,83 |
| PER | 80,96 | 86,48 | 83,59 |
| DAT | 88,69 | 87,22 | 87,87 |
| Overall | 76,61 | 69,30 | 72,72 |

demanding, because the similarity have to be computed $|N| \cdot |V|$ times, where $|N|$ is number of words in corpus and $|V|$ is size of vocabulary. It was very surprising for us, that even results of this experiment (table 5) did not show any significant change.

Tables 2–5 show the performance of the classifier for particular NE classes. The results for numbers (see the NUM row) are influenced by the fact, that 2 parts of the data used for testing do not contain any numbers. This leads to 0 precision, undefined recall and 0 F-measure. The average is then the F-measure around 56, but by leaving this two parts it is increased the value to around 75.

Apparently, for dates and times (DAT), persons (PER), numbers (NUM) and locations (LOC) the results are satisfactory in comparison to organizations (ORG) and other NEs (OTH). The results of organizations and other NEs are worse, because these classes are open and not well defined. The names of organizations are fuzzy and often can be distinguished from other classes only from the context or by using global knowledge (e.g. Johnie Walker). The NEs class “other” is even worse already from the definition since it covers a very wide range of entities. Entities from these classes are also not very often repeated in the corpus.

## 7 Conclusion and Future Work

We have created a new NE recognizer based on maximum entropy. Our recognizer achieved 76.78 recall, 69.58 precision and 72.94 F-measure and according to our best knowledge outperformed the previously published results. We have also conducted tests with semantic spaces used as a feature for our classifier. Our tests have so far shown no statistically significant improvement.

Our plan is to continue in the development of NER systems. We still believe in semantic spaces however a more proper way to use them is probably necessary. The combination of classifiers may also be a way to improve the results. At the moment a new corpus for Czech named entity recognition is being prepared. The corpus is designed to avoid some problems found in the Czech Named Entity corpus [8].

**Acknowledgments.** This research was supported by Advanced Computer and Information Systems project no. SGS-2010-028.

## References

1. Curran, J.R., Clark, S.: Language independent ner using a maximum entropy tagger. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 164–167. Association for Computational Linguistics, Morristown (2003)
2. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
3. Della Pietra, S., Della Pietra, V., Lafferty, J.: Inducing features of random fields. IEEE Trans. Pattern Anal. Mach. Intell. 19, 380–393 (1997)
4. Grishman, R., Sundheim, B.: Message understanding conference-6: a brief history. In: Proceedings of the 16th Conference on Computational Linguistics, COLING 1996, vol. 1, pp. 466–471. Association for Computational Linguistics, Stroudsburg (1996)
5. Isozaki, H., Kazawa, H.: Efficient support vector classifiers for named entity recognition. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, pp. 1–7. Association for Computational Linguistics, Morristown (2002)
6. Jones, M.N., Mewhort, D.J.K.: Representing word meaning and order information in a composite holographic lexicon. Psychological Review 114, 1–37 (2007)
7. Kozareva, Z., Ferrández, O., Montoyo, A., Muñoz, R., Suárez, A., Gómez, J.: Combining data-driven systems for improving named entity recognition. Data Knowl. Eng. 61, 449–466 (2007)
8. Kravalová, J., Žabokrtský, Z.: Czech named entity corpus and svm-based recognizer. In: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, NEWS 2009, pp. 194–201. Association for Computational Linguistics, Stroudsburg (2009)
9. Lund, K., Burgess, C.: Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods Instruments and Computers 28(2), 203–208 (1996)
10. Malouf, R.: A comparison of algorithms for maximum entropy parameter estimation. In: Proceedings of the 6th Conference on Natural Language Learning, COLING 2002, vol. 20, pp. 1–7. Association for Computational Linguistics, Stroudsburg (2002)
11. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, vol. 4, pp. 188–191. Association for Computational Linguistics, Morristown (2003)
12. Nocedal, J.: Updating Quasi-Newton Matrices with Limited Storage. Mathematics of Computation 35(151), 773–782 (1980)
13. Rohde, D.L.T., Gonnerman, L.M., Plaut, D.C.: An improved method for deriving word meaning from lexical co-occurrence. Cognitive Psychology 7, 573–605 (2004)

14. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, CONLL 2003, vol. 4, pp. 142–147. Association for Computational Linguistics, Stroudsburg (2003)
15. Ševčíková, M., Žabokrtsky, Z., Krůza, O.: Named entities in czech: annotating data and developing ne tagger. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 188–195. Springer, Heidelberg (2007)
16. Zhou, G., Su, J.: Named entity recognition using an hmm-based chunk tagger. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL 2002, pp. 473–480. Association for Computational Linguistics, Morristown (2002)

# Mining Significant Words from Customer Opinions Written in Different Natural Languages

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.** Opinions expressed by text documents freely written in various natural languages represent a valuable source of knowledge that is hidden in large datasets. The presented research describes a text mining-method how to discover words that are significant for expressing different opinions (positive and negative). The method applies a simple but unified data pre-processing for all languages, providing the bag-of-words with words represented by their frequencies in the data. Then, the frequencies are used by the algorithm which generates decision trees. The tree decisive nodes contain the words that are significant for expressing the opinions. Positions of these words in the tree represent their significance degree, where the most significant word is in the node. As a result, a list of relevant words can be used for creating a dictionary containing only relevant information. The described method was tested using very large sets of customers' reviews concerning the on-line hotel room booking. For more than 15 languages, there were available several millions of reviews. The resulting dictionaries included only about 200 significant words.

**Keywords:** textual documents, multilingual documents, natural language, opinion analysis, text mining, decision tree, significant attribute.

## 1 Introduction

The research goal was to find a method how to create a dictionary that contains only words significant for expressing opinions. One of typical electronic text utilizations is collecting and analyzing expressions of opinions provided by people that used some services or purchased some goods. The more expressions are available, the more valuable information and knowledge can be revealed inside, which can be later used both in commercial and non-commercial areas [9]. On the other hand, very large volumes of data need processing by machines, and, in addition, data having the form of unstructured natural language documents are generally difficult for such processing [1]. Further, the problem is intensified when customers use many different languages because there is not an unified method developed for the easy processing of such data. Particularly the commercial branch is very interested in discovering knowledge within the data that are usually systematically collected for a long time, sometimes tens of years [5]. Customers express their opinions which can be positive or negative, or which can be graded using a certain scale [10].

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 211–218, 2011.

One of questions can be: *What is significant for including a certain opinion into one of categories like satisfied or dissatisfied customers?* In other words, what attributes are relevant? Obviously, the answer is hidden in the words and phrases, and how they are used. The dictionary should not be too large, however, it is not easy to assess its size in advance. Human beings can more or less reliably and easily recognize the meaning of a presented opinion, however, if the opinions are written in many – more than two or three – foreign languages, the task becomes very difficult and not too many people know more languages sufficiently. Contemporary, a lot of institutions and business organizations are engaged in a trade at many countries and continents, and the problem with processing the large heterogenous data volumes is now very current.

## 2 Data Description

The text data used in the experiments contained opinions in many languages of several millions customers who – via the on-line Internet service – booked accommodations in many different hotels and countries. The on-line hotel reservation web has the hotels organized in a hierarchy *continent-country-city-(city district)-hotel*. Besides the information about the hotel prices, facilities, policies, terms, and conditions, the web contains user reviews related to their stay in a given hotel. The reviews cannot be entered by any person but only by the people that made a reservation through the web and stayed in the hotel. Each review consists of identification of the reviewer, his or her overall evaluation (a number on a 10-point scale) and the review text. The identification includes the type of the customer (solo traveller, family with young children, mature couple, and so like), the country and city where the reviewer comes from, and the date of the review. The review texts have two parts – a negative and positive experience with the hotel, both written in a natural language. The data was collected from more than 100 countries, 25,000 cities and districts, 108,000 hotels, and contained about 5,000,000 reviews written in many languages.

Such a big number of labeled examples enables to create sufficient training sets for an interesting number of different languages. Because of the policies of the hotel reservation web, the samples are labeled as positive and negative relatively carefully. On one hand, they are often written quite formally; on the other hand, most of them embody all deficiencies typical for texts written in natural languages (mistypings, transposed letters, missing letters, grammar errors, sometimes combinations of two languages in one item, and so on). Often, languages that normally use diacritic (for example, Czech, French, Spanish, and others) are used sometimes with and sometimes without it (written in the plain ASCII/ANSI code-page).

Labeling the reviews by the country of the reviewer enabled to automatically extract texts originated in different countries. However, belonging to a particular country doesn't necessarily mean that the reviewer used the language of that country. This means that, for instance, some reviews written by Czechs were written also in English, Slovak, or German or in combination of more languages. Just one copy of an illustrative example, Czech-English mixture, with all original mistakes: '*Co se mi nelíbylo byl hluk z ulice. There was litlle bit noise from street.*'

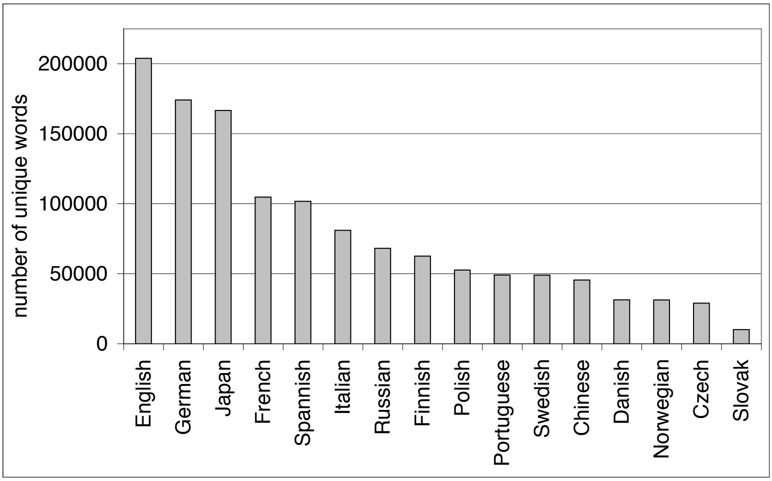


**Fig. 1.** Number of unique words for processed data sets

In addition, a reader of those reviews can see not only Latin alphabet: Chinese, Hebrew, Japanese, Korean, Russian, Serbian, Thai, and others. This introduces additional difficulties because each such language has its own specifics [6]. For example, Japanese texts used all three main scripts: *Kanji* (individual characters) and syllabic *Hiragana and Katakana* (in addition, they also contained sentences written in English). Another difficulty related to automatic processing of Chinese and Japanese text data is that, unlike many other languages, they do not have explicit whitespace between words. As a result, a special form of word segmentation, which is a difficult problem in these languages, is normally required before further processing [7]. Here, the authors used the same method for all languages and consequently sometimes, instead of strictly individual words, certain phrases were separated. Using a simple approach described further, the list of important words in Japanese contained, for example, 'words' such as *narrow room*, *small room*, or *since there is no elevator*.

Altogether, up to this time, there were processed the following languages: *Croatian, Czech, Danish, Finish, French, German, Hebrew, Italian, Japanese, Korean, Lithuanian, Norwegian, Polish, Portuguese, Russian, Slovak, Spanish, Swedish, Thai,* and *Chinese*. Some other data were not processed up to now because they are, for example, *Spanish*, but from *Argentina* and maybe from other South American countries. Similarly *Brazilian Portuguese*. Also, *Bassa, Catalan, Dutch, Esperanto, Estonian, Hungarian, Lithuanian, Romanian, Slovene*, and *Turkish* were not processed up to now, too, because their number of samples was too small. As a rarity, one positive review is written in *Avestan*, which is an archaic East Iranian (probably dead) language.

## 3 Text Document Pre-processing and Representation

The initial pre-processing of all languages was very simple and standard, based only on creating a *bag-of-words* and, consecutively, a *dictionary* from words in text documents available [8]. The pre-processing did not use removing *stop-words* because up to now, due to many different languages and providing the same conditions to each of them, there was not enough time to create lists of such words – this is a task for the near future. The words were finally represented simply by their frequencies because other possibilities (for example, *TF*x*IDF)* did not bring any advantages as the preliminary experiments showed. The dictionary (as well as each document) was transformed into a multidimensional vector where individual word frequencies were used as coordinates within the abstract space with each dictionary word as one of dimensions. Typically, the individual vectors were very sparse because of the very large dictionary for every language and small number (in order of tens) of words in each review. For example, the collection of ca 17,000 Czech texts had about 29,000 unique words, 57,000 Russian texts 68,000 unique words, 356,000 Italian texts 81,000 unique words, 470,000 Spanish texts 102,000 unique words, and the largest collection of 1,919,000 English texts 204,000 unique words.

**Table 1.** The most significant words for Czech and Russian

| **Czech** | | | **Russian** | | |
|---|---|---|---|---|---|
| Original | % | Translation | Original | % | Translation |
| špatně | 100 | wrong | расположение | 100 | location |
| nedostatečné | 100 | insufficient | хороший | 88 | good |
| nemožnost | 99 | impossibility | отличный | 81 | excellent |
| nedostatečná | 99 | insufficient | приветливый | 78 | friendly |
| chyběl | 99 | missing | уютный | 75 | comfortable |
| chybějící | 98 | missing | отличное | 73 | excellent |
| koberec | 98 | carpet | доброжелательный | 71 | friendly |
| stěny | 98 | walls | отзывчивый | 70 | responsive |
| netekla | 98 | no water | месторасположение | 69 | location |
| zápach | 97 | smell | прекрасный | 67 | beautiful |
| nelíbila | 97 | dislike | тихое | 66 | quiet |
| nepříjemné | 97 | unpleasant | доброжелательность | 66 | generosity |
| nedostatečně | 97 | insufficiently | просторный | 65 | spacious |
| nefungoval | 96 | not working | природа | 65 | nature |
| nefunkční | 96 | not working | приятный | 65 | pleasant |

The simply created bag-of-words suffered from obvious, commonly known shortages, for example, containing several variants of the same word – the experiments, however, had no available batch-mode stemming tools for most of all languages. Therefore, the authors decided to use all word forms because here the goal of experiments was *not to reach the best possible classification*, which is often an intention. The dimensionality of dictionaries also increases for languages that use diacritical marks (accents). Some reviewers often omitted these symbols and therefore both versions of the same word appeared.

## 4 Creating Dictionaries of Significant Words

For each processed language, the main goal was building a dictionary containing the words that were significant for expressing the *positive* or *negative* opinion, as well as to reveal how each word contributed to the positivity or negativity. Knowing the labels of the available samples, the authors decided to employ a decision tree generator [2] which constructs a classifier that can separate positive and negative reviews. Decision trees typically use only part of attributes and represent a set of rules. Having a set (dictionary) of a huge number of attributes (words), only a fraction of them can be decisive. One of the most popular tree generators is the algorithm that builds a tree using minimization of entropy – that is, recursively splitting the original heterogenous set of samples into homogeneous subsets. The splitting is driven by those attributes that provide the highest entropy decrease. Here, such attributes are words that direct the labeled samples to leaves as homogeneous as possible (subsets containing, if possible, only items from one class). The entropy $H(X)$ of a discrete random variable $X$ with possible values $\{x_1, \ldots, x_n\}$ is defined by the following equation:

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_b p(x_i). \tag{1}$$

If $p(x_i) = 1$ or 0, the entropy is zero, which means that a subset contains items only from one class and no items from other classes. If (for two classes) the items in a subset are divided $50\%$ : $50\%$, the entropy is maximal. The most significant attribute is in the tree root, other important ones on levels close to the root. Therefore, the position of a word in the tree determines its significance. The tree asks its root each time, so this word has the $100\%$ importance. Other words on the following tree levels may gradually be less and less important from the splitting point of view because the classification does not ask them each time, depending on a selected branch. Naturally, the classification accuracy must be as high as possible, otherwise the decisive words in the tree nodes cannot be selected reliably. In this way, the classification accuracy plays also an important role and a big number of labeled training samples is necessary. After building the tree, all significant words are known and the dictionary with words weighted by their significance can be created. Such words are then rated to be important for expressing the opinion, while the rest of them plays a negligible or no role and from the statistical viewpoint represents noise. The dictionary size (its number of unique words) is given by the tree size (its number of interior, decisive nodes). As a software tool, the authors applied the efficient c5/See5 decision tree [3].

## 5 Experiments and Their Results

After the initial data pre-processing, it turned out that for most of the languages the data volume was too large to be processed because of the memory requirements. The authors decided to split the primary datasets into smaller subsets using a random selection of 50,000 samples per each subset. In addition, removing words with frequency $< 2$ decreased the dimensionality (experiments showed that even after this filtering the results

**Table 2.** The most significant words for Italian and Spanish

| Italian | | | Spanish | | |
|---|---|---|---|---|---|
| Original | % | Translation | Original | % | Translation |
| mancanza | 100 | lack | no | 100 | not |
| non | 97 | not | demasiado | 76 | too |
| scarsa | 95 | poor | falta | 76 | absence |
| troppo | 79 | too | poco | 75 | bit |
| assenza | 78 | absence | olor | 70 | smell |
| carente | 75 | lacking | oía | 70 | heard |
| rumorosa | 75 | heavy | escasa | 70 | insufficient |
| odore | 72 | odor | oye | 69 | hear |
| poca | 71 | little | mala | 69 | bad |
| po | 71 | bit | dificultad | 69 | difficulty |
| sarebbe | 70 | would | excesivo | 69 | excessive |
| posizione | 70 | location | debería | 69 | I should |
| pò | 68 | bit | ruido | 69 | noise |
| rumorosità | 67 | noise | mal | 68 | bad |
| poco | 66 | little | escaso | 68 | insufficient |

**Table 3.** The most significant words for German and French

| German | | | French | | |
|---|---|---|---|---|---|
| Original | % | Translation | Original | % | Translation |
| lage | 100 | location | calme | 100 | quiet |
| gutes | 78 | good | pas | 89 | not |
| nicht | 76 | no | trop | 72 | too |
| freundliche | 74 | friendly | bon | 70 | good |
| freundliches | 73 | friendly | manque | 69 | absence |
| freundlich | 70 | friendly | mal | 67 | wrong |
| nettes | 65 | nice/kind | odeur | 66 | smell |
| nette | 64 | nice/kind | peu | 66 | little |
| zentral | 62 | central | absence | 66 | absence |
| gute | 61 | good | mauvaise | 61 | bad |
| schönes | 61 | nice | insuffisante | 60 | insufficient |
| freundlichkeit | 59 | friendliness | bruyant | 60 | noisy |
| zuvorkommend | 56 | helpful | plastique | 60 | plastic |
| tolles | 56 | great | excessif | 59 | excessive |
| ruhig | 55 | quiet | bruit | 59 | noise |

were quite identical). Then, the decision tree was applied to all individual (sub)sets of data and the trained trees were used for creating the dictionaries of words significant for expressing the opinions. Which words belong to the positive or negative class, it is given by a relevant branch which terminates in a 'positive' or 'negative' leaf. The words were represented by their frequencies. The decision tree mostly asked if the frequency was $>0$ or $=0$, which was, in fact, the binary representation. However, sometimes the

**Table 4.** The most significant words for Japanese and English

| Japanese | | | English | |
|---|---|---|---|---|
| Original | % | Translation | Original | % |
| 部屋が狭く | 100 | narrow room | location | 100 |
| 部屋が狭い | 100 | the small room | friendly | 80 |
| エレベーターがないので | 100 | since there is no elevator | not | 77 |
| 強いて言えば | 100 | if I'm forced to say | excellent | 73 |
| 結局 | 100 | eventually | helpful | 67 |
| 残念 | 99 | shame | spacious | 62 |
| 残念でした | 99 | too bad | friendliness | 59 |
| 清潔 | 99 | clean | beautiful | 57 |
| 駅から近く | 99 | near station | comfortable | 55 |
| 便利でした | 99 | was useful | nice | 55 |
| スタッフも親切でした | 99 | the staff was kind | conveniently | 54 |
| 部屋は清潔 | 99 | rooms are clean | convenient | 53 |
| 狭い | 98 | narrow | fantastic | 53 |
| 快適に過ごせました | 98 | we had a comfortable | good | 52 |
| ロケーション | 98 | location | proximity | 52 |

real frequency value played its role, therefore the results were slightly worse for the binary encoding.

Each of the 50,000-samples subsets gave almost the same list of words; usually, only the position of the words fluctuated. The number of selected significant words was only around 200. In the tables Tab. 1 – Tab. 4, because of the limited article size, only the first 15 words with their average position weight (column %) is shown for selected languages. As for reliability of this selection, the accuracy of the classifier was typically between 85-93% – lower values for lower numbers of training samples depending on a language. The accuracy estimate of the classifier performance was given by the 5-times 10-fold-crosvalidation procedure. To avoid the classifier overfitting, the c5/See5 default global pruning confidence factor 25% was applied.

## 6 Conclusions

This research describes a method of extracting significant words from unstructured textual customer reviews written in various natural languages. The word significance was given by expressed opinions of provided services (hotel accommodations), positive and negative. Because of the very large data volumes in many different languages, the suggested approach chose a relatively simple but unified way to pre-process the data. The application of the c5/See5 decision tree generator selected the significant words according to their belonging to the positive or negative opinion (the classification itself was not the goal). At the same time, the position of the tree nodes (that contained the words) represented the significance degree of the individual extracted decisive words – the most important one was in the root. This approach noticeably decreased the number of all words (in the order of $10^4$ to $10^5$) to about 200. Such an approach can be used for building dictionaries that include only relevant information and are not too large. The dictionaries then enable more effective analysis of the data, which is attractive both for commercial and non-commercial entities. Selecting significant words, as described above, was succesfully applied to building dictionaries with words expressing opinions in natural languages [4]. In addition, those dictionaries were further used for looking for significant phrases (submitted to another conference). The future research is going to focus on further improvement of the suggested approach, mainly on the data preprocessing phase, which is complicated by the large language variety and high data volumes. Another part will include more detailed opinion analysis when the reviews can be graded, having more than two classes.

**Acknowledgements.** The research work published in this paper was supported by the Research program of Czech Ministry of Education VZ MSM 6215648904. The authors thank Eva Dařenová and Eva Miškovičová for their valuable help with analysing text documents written in several foreign languages.

## References

1. Berry, M.W., Kogan, J. (eds.): Text Mining: Applications and Theory. John Wiley & Sons, Chichester (2010)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2007)
3. c5/See5 (2011), http://www.rulequest.com/see5-info.html
4. Dařena, F., Žižka, J.: Text Mining-Based Formation of Dictionaries Expressing Opinions in Natural Languages. In: Proceedings of the 17th International Conference on Soft Computing Mendel 2011, Brno, June 15-17, pp. 374–381 (2011) ISSN: 1803-3814
5. Liu, B.: Web data mining: Exploring Hyperlinks, Contents, and Usage Data. In: Opinion Mining. Springer, Heidelberg (2006)
6. Nie, J.Y.: Cross-Language Information Retrieval. Synthesis Lectures on Human Language Technologies 3(1), 1–125 (2010)
7. Peng, F., Huang, X.: Machine learning for Asian language text classiffication. Journal of Documentation 63(3), 378–397 (2007)
8. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 1, 1–47 (2002)
9. Shmueli, G., Patel, N.R., Bruce, P.C.: Data Mining for Business Intelligence. John Wiley & Sons, Chichester (2010)
10. Žižka, J., Dařena, F.: Automatic Sentiment Analysis Using the Textual Pattern Content Similarity in Natural Language. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 224–231. Springer, Heidelberg (2010)

# On Positive and Unlabeled Learning for Text Classification*

István Nagy T.[1], Richárd Farkas[2], and János Csirik[3]

[1] University of Szeged, Department of Informatics,
6720 Szeged, Árpád tér 2., Hungary
[2] Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung,
Azenbergstrasse 12, D-70174 Stuttgart, Germany
[3] MTA-SZTE Research Group on Artificial Intelligence,
6720 Szeged, Tisza Lajos krt. 103., Hungary
{nistvan,jcsirik}@inf.u-szeged.hu,
farkas@ims.uni-stuttgart.de

**Abstract.** In this paper we present a slightly modified machine learning approach for text classification working exclusively from positive and unlabeled samples. Our method can assure that the positive class is not underrepresented during the iterative training process and it can achieve 30% better F-value when the amount of positive examples is low.

**Keywords:** semi-supervised learning, positive and unlabeled, PU, text classification.

## 1 Introduction

Classification is a well-studied problem in machine learning. Text classification is the process of assigning predefined category labels to new documents. The classic supervised approach to build a text classifier is to first manually label a set of training documents, which are labeled with the same set of predefined category or class labels as the test set. A classification algorithm is then applied to the training data to build a classifier, which is subsequently employed to assign the predefined classes to instances in the test set (for evaluation) or future instances (in practice).

The main bottleneck of supervised learning is that it needs a large number of labeled training examples for building the accurate model. However, labeling is typically done manually, which is – on the one hand – labor intensive and time consuming, and – on the other hand – may lead to unexpected complications. For example, in practice some of the test or future instances may not belong to any of the predefined classes of the original training set. Furthermore, the test set may contain additional unknown subclasses, or new subclasses may arise as the underlying domain evolves over time. Manual annotation cannot be effectively prepared for these cases. Collecting negative training examples is especially delicate and arduous because negative training

* This work was supported in part by the National Innovation Office of the Hungarian government within the framework of the projects BELAMI and MASZEKER.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 219–226, 2011.

examples must be uniformly represented in the universal set excluding the positive class. On the other hand, manually collected negative training examples could be biased because of humans' unintentional prejudice, which could be detrimental to classification accuracy.

In recent years, researchers have studied the concept of using only a small labeled set and a large unlabeled set to help learning. This approach is called semi-supervised learning. It reduces the effort of manual labeling, but negative examples are also needed. The Positive and Unlabeled (PU) learning is a specific semi-supervised learning method. PU learning approaches only need a positive set P and an unlabelled set U, then the algorithm can identify hidden positive documents in the unlabeled set.

Although there are several PU implementations, we were motivated to investigate them in detail because many real life problems can be successfully solved by PU approaches. For example, a company that tries to attract new customers with direct marketing owns a database of their own customers, but has no information on those who are not their customers. In this case, they only have positive examples but no negative examples. If they buy a database containing data on people, they can find people who are similar to their customers, and then can deliberately seek out the bids. In this paper we focus on the test classification use case which is usually used as a sandbox for PU algorithms.

## 2 Related Work

Traditionally, PU learning algorithms are based on a strategy of two steps: first, identify a set of reliable negative (RN) documents from the unlabeled set by using any method for this. Second, build a classifier based on positive and reliable negative data from the unlabeled set. The specific difference between the various algorithms in these two steps is as follows: the 1-DNF or PEBL algorithm [11] first collects words that occur in the positive set P more frequently than in the unlabelled set U. This method has several versions with some modifications. One of them improved the original 1-DNF algorithm [12], which requires the absolute frequency of the feature in the positive data set greater than $\alpha\%$. As a result, they had a smaller but better quality positive word set, which yields a much larger reliable negative document set. The other popular approach to identify reliable negative examples in U is the Spy technique [4]. It is based on the method that first randomly selects a set S of positive documents from P and puts them in U. These documents are called "Spies". They behave similarly to the unknown documents in U. Hence, the algorithm can identify the behavior of the unknown positive documents in U more easily. The Rocchio algorithm is also a popular text classificaton method [4,2]. In this technique, reliable negative documents are extracted by using the information retrieval method Rocchio. This approach constructs a prototype vector for every class. The classifier is then used to classify documents in U. Those documents that are classified as negative are considered (reliable) negative data, denoted by RN. In the second step, a classifier is built using expectation-maximization (EM) or Support vector machine (SVM) iteratively.

# 3 The Proposed Technique

The most popular PU learning algorithms apply a common two-step strategy. We examined the two steps separately. The key element in the first step is the quality of reliable negative documents. To investigate this, we used error rate as follows [12]:

$$Err(\%) = \frac{\#positive\_examples\_in\_RN}{\#positive\_examples\_in\_U}$$

We found that several approaches are good at identifying reliable negative examples from the unlabelled data set. This is proved by the fact that the error rate was less than $1\%$ when we applied Rocchio. Thus, the second step seemed to be more interesting to explore. In the common second step SVM or EM machine learning algorithms were used. Basically, we investigated the iterative SVM method. In the basic iterative SVM method, P and RN were first used to train SVM. Let Q be the remaining unlabeled document set: Q = U - RN. In each iteration we used the actual SVM model to classify Q. Documents which the model classifies as negative were put in RN. The main idea of this approach is that SVM can extract a greater number of possible negative examples in Q. If the current model could not mark any more documents from Q as negative, the algorithm terminated. After the iterations, the final classifier is applied. However, sometimes it proves to be necessary to select a classifier since SVM is sensitive to noise. It may happen that an iteration of SVM extracts too many positive documents from Q and puts them in RN, which may have a negative effect on the performance of the last SVM classifier.

```
while true
 use P and RN to train SVM classifier;
 classify Q using SVM model;
 let W be the set of documents that a current SVM model
   classified as negative;
 if W = $\emptyset$ then
  exit;
 else
  Q = Q - W;
  RN = RN $\cup$ W
```

**Fig. 1.** The two-step approach of [11]

However, we wanted to see how other classification algorithms can perform. In all cases, Rocchio was applied as the first step. In most cases, this approach could identify a reliable negative set which is large enough with minimum error rate.

### 3.1 The Rocchio Technique

In Rocchio classification, each document is represented as a vector in [7]. Each element in the vector was weighted in term frequency-inverse document frequency (tf-idf). This weight measures how important a word is to a document in a corpus. A classifier is built by constructing positive and negative prototype vectors. In classification, for each test document, it simply uses the cosine measure [7] to compute the similarity of the test document to each prototype vector. The class whose prototype vector is most similar to the test document is assigned to the test document. Documents classified as negative form the negative set RN.

### 3.2 The Proposed Technique

As [3] emphasized, catching the best iteration during the iteration running process is an important question. The current systems increase the size of only the RN set in the second step. In this case, since only a small positive set is used, in the last iterations the RN set becomes much larger than P, yielding that the positive class is underrepresented and the negative class is overrepresented. Thus, the model may classify all examples as negative, or learn the negative class. To solve this, we increased the P set too. Since the algorithms that we applied could determine the example class probability, they just accepted the prediction in case the prediction probability was higher than $\alpha$ value. Finally we used $\alpha$ = 0.9. As shown in Figure 2 during each iteration, both the P and the RN class were increased.

```
while true
 use P and RN to train classifier;
 classify Q using a model;
 let W be the set of documents that a current model
   classified as negative;
 let S be the set of documents that a current model
   classified as positive;
 if $W = \emptyset$ then
  exit;
 else
  $Q = Q - W - S$;
  $RN = RN \cup W$;
  $P = P \cup S$;
```

**Fig. 2.** The modified iterative method

### Classifiers

In our experiments we used the implementations available in the WEKA [10] library, an open-source data mining software written in Java.

**Boosting** is [8] a way of improving the performance of a weak learning algorithm. The algorithm first generates a set of classifiers of the same type by applying bootstrapping on the original training data set. These classifiers vote a decision. The final decision is made using a weighted voting schema for each classifier. The resulting model is many times more accurate than the original. Some iterations of Boosting were performed on each model. Further iterations gave only slight improvement in the F-measure (less than 0.05%), thus we decided to perform only 10 iterations in each experiment.

**C4.5** is based on the well-known ID3 tree learning algorithm [6]. Axis-parallel hyperplanes are used in classification, and hence learning is very fast. We built decision trees that had at least 2 instances per leaf, and used pruning with subtree raising and a confidence factor of 0.25.

**Support Vector Machines (SVM)** [9] is the linear function of the form $f(x) = w^t x + b$. Between the weight vector w and the input vector x, $w^t x$ denotes the inner products. SVM is based on the main idea of selecting the hyperplane that separates the space (between the positive and negative classes) while maximizing the smallest margin. In practice we used libSvm[1] and the Weka **SMO** implementation.

**Logistic Regression** is a predictive model. It was used when the target variable is a categorical variable with two categories. The logistic model formula computes the probability of the selected response as a function of the values of the predictor variables.

## 4 Experiments and Results

In our experiments, we used the Reuters-21,578 dataset, which has 21,578 documents collected from the Reuters newswire, as our training and testing sample set. Of the 135 categories in Reuters-21,758, only the 10 most frequent ones are used. PU approaches were evaluated in different ways. On the one hand, we used the inductive evaluation: the model identifies or retrieves positive documents from the unlabeled set U. On the other hand, the model was evaluated on the test set with unknown examples. In both evaluation methods one category was employed as the positive class, and the other nine classes as the negative and each category fulfilled once the role of the positive class. For each category, 30% of the documents were randomly selected as test documents. The rest was used to create the training sets as follows: $\gamma$ of the documents from the positive class are first selected as the positive set P. The rest of the positive documents (1-$\gamma$) and

**Table 1.** The modified second step with different learning algorithms on the Reuters-21,578 dataset (F-value results). Roc-SVM: the online available PU learning algorithm, SVM: during the iteration process in the modified second step, libSVM was used as learning algorithm. SVM_B: boosted version at SVM. C4.5: decision tree was used in the second step. C4.5_B: boosted version at C4.5. LogReg: Logistic Regression was used in the second step. LogReg_B: boosted version at Logistic Regression. SMO: SVM implementation in Weka was used in the second step. SMO_B: boosted version at SMO.

| **p%** | **Roc-SVM** | **SVM** | **SVM_B** | **C4.5** | **C4.5_B** | **LogReg** | **LogReg_B** | **SMO** | **SMO_B** |
|---|---|---|---|---|---|---|---|---|---|
| 0.05 | **0.30195** | 0.0707 | 0.4130 | 0.5017 | **0.6591** | 0.4084 | 0.4034 | 0.6141 | 0.6280 |
| 0.1 | **0.68731** | 0.3727 | 0.6641 | 0.7029 | 0.7463 | 0.4187 | 0.4062 | **0.7611** | 0.7600 |
| 0.15 | **0.76898** | 0.5311 | 0.7583 | 0.7300 | **0.8089** | 0.4082 | 0.4453 | 0.7941 | 0.7936 |
| L 0.2 | **0.79846** | 0.7014 | 0.7727 | 0.7871 | **0.8372** | 0.4431 | 0.4336 | 0.7892 | 0.7911 |
| 0.3 | **0.82053** | 0.7932 | **0.8145** | 0.8027 | 0.8052 | 0.4486 | 0.4449 | 0.8072 | 0.8038 |
| 0.4 | **0.8314** | 0.8271 | **0.8343** | 0.8219 | 0.8228 | 0.4657 | 0.4600 | 0.8157 | 0.8161 |
| 0.45 | **0.82432** | 0.8222 | **0.8460** | 0.8012 | 0.8174 | 0.4935 | 0.4633 | 0.8074 | 0.7989 |
| 0.5 | **0.80254** | 0.8188 | 0.8198 | 0.8213 | **0.8294** | 0.5246 | 0.5262 | 0.7841 | 0.7864 |

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

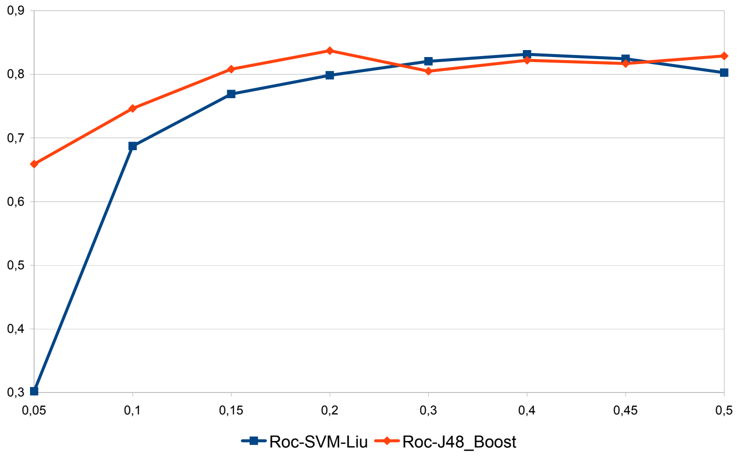


**Fig. 3.** The result of Roc-SVM and our modified second step with boosted C4.5 learning algorithm in inductive evaluation

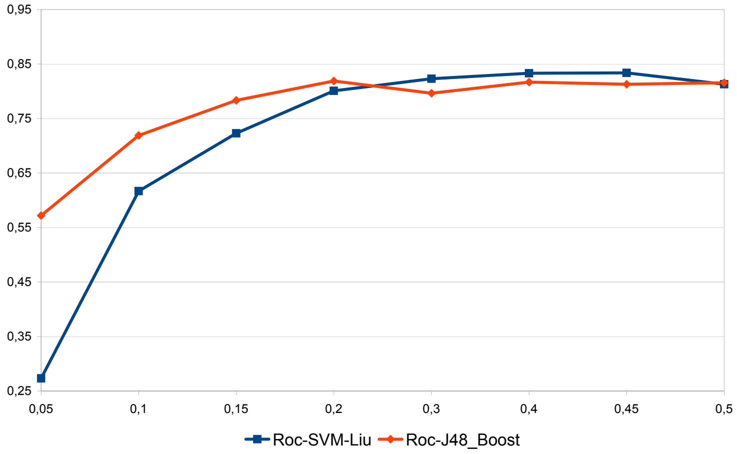


**Fig. 4.** Evaluating on a separate test set of Roc-SVM and our modified second step with boosted C4.5 learning algorithm

negative documents are used as the unlabeled set U. In the inductive evaluation method the test set was added to U. Since we would like to compare our results with [2] and [4] we determined the values of $\gamma$ as 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.45 and 0.5. In order to compare our results with existing methods, we evaluated the Roc-SVM approach too. It is available on the Web as part of the LPU system[2]. To evaluate the performance of the classifier we used F-value. It is the harmonic mean of precision and recall.

Table 1 shows the result achieved by different learning algorithms. As the table shows, in the case of the libSVM and the C4.5 algorithms, boosting was able to improve the results. However, in the case of SMO and LogReg, boosting was not effective. Since the boosted C4.5 algorithm achieved the best results in low $\gamma$ rate, we compare our modified second step method with this learning algorithm with the ROC-SVM. So, we evaluate and compare these two methods in two different ways. Results are shown in Figures 3 and 4.

As the figures show, the boosted C4.5 algorithm with the modified second step achieved better results when $\gamma$ was low. The biggest difference between the two approaches could be observed when the positive rate was the lowest (0.05). In this case our method could achieve an F-value about 30% better than Roc-SVM. As the positive rate grows, this advantage declines. If the rate of the positive elements is higher than 30%, the two approaches perform nearly identically.

## 5 Conclusion

The common PU algorithms are based on a two step strategy. We found that the Rocchio approach could effectively solve the first step. Therefore, we investigated the second step. To improve the iterations we modified the common second step: if the learning algorithm classifies an unlabeled element as positive we put it the positive set P, yielding that the positive class is not underrepresented during the iteration process. The common PU learning algorithms used SVM or EM iteratively. We examined some other learning algorithms, and we found that the boosted C4.5 learning approach achieved more than 30% better F-value when the positive rate was low.

## References

1. Agresti, A.: Building and applying logistic regression models. In: An Introduction to Categorical Data Analysis, pp. 137–172. Wiley, Chichester (2007)
2. Li, X., Liu, B.: Learning to classify text using positive and unlabeled data. In: Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico, (August 9-15, 2003)
3. Li, X., Liu, B., Ng, S.-K.: Negative Training Data can be Harmful to Text Classification. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-10). MIT, Massachusetts (2010)
4. Liu, B., Dai, Y., Li, X., Lee, W., Yu, P.: Building text classifiers using positive and unlabeled examples. In: Proceedings of the Third IEEE International Conference on Data Mining (ICDM 2003), Melbourne, Florida (November 19-22, 2003)

[2] http://www.cs.uic.edu/~liub/LPU/LPU-download.html

5. Liu, B., Li, X., Lee, W.S., Yu, P.S.: Text Classification by Labeling Words. In: AAAI 2004 (2004)
6. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Francisco (1993)
7. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
8. Schapire, R.E.: The Strength of Weak Learnability. Machine Learning 5(2), 197–227 (1990)
9. Vapnik, V.N.: Statistical Learning Theory. Wiley, Chichester (1998)
10. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)
11. Yu, H., Han, J., Chang, K.C.C.: PEBL: Positive Example-Based learning for web page classification using SVM. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 239–248. ACM, New York (2002)
12. Yu, H., Zuo, W., Peng, T.: A New PU Learning Algorithm for Text Classification. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 824–832. Springer, Heidelberg (2005)

# Optimisation Approach to the Construction of the Polish Morphological Guesser

Michał Jastrzębski

University of Warsaw,
Faculty of Mathematics, Informatics and Mechanics
m.jastrzebski@students.mimuw.edu.pl

**Abstract.** This document presents a method of constructing morphological guesser for Polish language (a tool for guessing morphological descriptions of words not known to the dictionary) based on a method of choosing an optimal guessing strategy when applied to the training set. This involves tackling with an exponential growth of the number of strategies to test and maximization of a certain multivariable function.

**Keywords:** morphological guesser, tagger, optimization.

## 1 Introduction

Morphological dictionaries, given a word form, assign to it the set of all possible tags it may possess, based only on the form, regardless the context. Resulting sets of tags are then used by taggers, which given the whole text try to disambiguate each word by finding the most suitable tag for its usage in the given context. Unfortunately, one may encounter word forms which the morphological dictionary cannot recognise. One obvious solution would be to assign all tags from the tagset to each of these words and allow the tagger to perform the full disambiguation. Note that this may influence the results of tagging the other words in the text. Alternative approach is to construct a morphological guesser which would act as a morphological dictionary. This means that receiving an unknown word form, the system would guess the set of tags it may possess, based solely on its form and, of course, based on the morphological dictionary results for words it recognises.

For a highly inflectional language such as Polish, one should expect that tags assigned to a given word can be easily determined by its suffix. Piasecki and Radziszewski in [1] showed a simple approach to morphological guessing based only on words endings. Their approach is based on an index tree constructed from suffixes. Here, we will also use a similar structure (described in the following subsections). The approach we take is more complex and general. We will define a notion of a *guessing function* which based on the word form produces the corresponding set of tags. Each of these functions defines a different guessing algorithm which we apply to the training set consisting of words recognized by the dictionary. Next, we will define a measure comparing the algorithms based on their results. The one presenting the best results will be chosen as a

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 227–234, 2011.

final guessing function in our algorithm. The main problem we encounter, is the number of guessing functions which grows exponentially which makes it impossible to test them all.

The main contribution of this paper is deriving an interesting algorithmical technique of choosing the best guessing function by reducing the problem size from exponential to polynomial. This technique will be described in details in following sections. From the methodological point of view, our approach is interesting as it may be used for other inflectional languages. What is more, note that all guessers using the rules such as *choose* $80\%$ *of most frequent tags of words ending with this suffix*, are the subcases of our general guessing function.

The last section shows some of the preliminary results of guesser implementation using our optimisation techniques.

## 2 General Approach

### 2.1 Constructing a Tree from Reversed Suffixes

At the very beginning the guesser needs to gather the data describing the knowledge of a morphological dictionary about known words. This consists of creating a tree containing reversed suffixes and adding morphological information to this structure using the dictionary.

Piasecki and Radziszewski in their paper give a detailed description of a tree construction. Here we'll describe it only briefly. Each word known to the dictionary is transformed to a lowercase string, then reversed and finally added to the tree letter by letter.

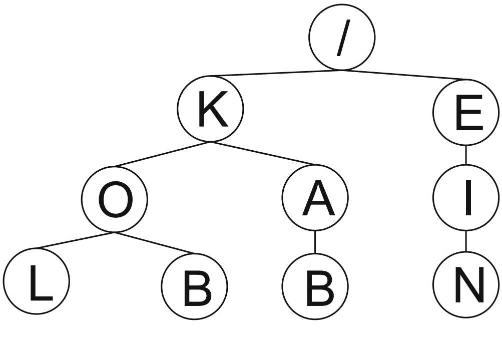


**Fig. 1.** A simple suffix tree build from Polish words *bak*, *bok*, *lok* and *nie*. / states for the empty string - the root of the tree

An example of a suffix tree is shown in Figure 1. Note that in the end the structure contains all words known to the dictionary. As we can see, the lowest common ancestor of two words in this tree is a node corresponding to the longest common suffix of these words. Having the tree constructed, we assign to each node some morphological information consisting of all tags of all known words ending with the suffix corresponding to this node. In our example, the node containing the letter *O* (corresponding to the suffix *KO*) would be assigned with all the tags assigned to words $lok$ or $bok$.

### 2.2 Introducing a Notion of a Guessing Function

Let us introduce a notion of a *guessing function*. This is, we define a function $S$ assigning to each tree node a set of tags. Formally, if $V$ is the set of tree nodes and $T$ is the set of all possible tags in the tagset, we say that a *guessing function* is any function satisfying

$$S : V \to 2^T.$$

This assignment $S$ is a basis of the guessing algorithm. The procedure starts by selecting a number $L$ which we will call the *cut-off length*. Next, for each word we are supposed to guess tags for, we do as following (let the Polish word *obok* be an example)

1. Reverse the word (*obok* → *kobo*).
2. Find the longest prefix of the word with length not greater than $L$, represented as a tree node (*kobo* → *ko*).
3. Return the set of tags associated with the node by our guessing function $S$ (return $S(ko)$).

From the description above we may see that if two word forms end with a common suffix of length greater than $L$, we guess that they share the same set of tags. The value of $L$ can be modified manually, but our initial tests have shown that the best results are obtained for $L = 6$. This value gives us a good trade-off between the amount of linguistic information contained in the suffix (the longer the better) and the amount of statistical data assigned to a particular node in the tree (for a long suffix, the number of words ending with it is too small).

### 2.3 Choosing the Right Function

The core idea behind the method introduced in this paper is applying algorithms defined by all possible guessing functions to the training set consisting of all the words known to the dictionary and choosing the one which will maximize a certain measure. We have chosen the *F-measure* as a value being maximized. Assume that we have set up a certain guessing function $S$ (and thus an algorithm defined by it). Unfortunately, it is impossible to directly run the guessing procedure on the words in the training set. Indeed, each of these words is already represented in the tree (as the tree has been built up from the training set). This means that for a word of length greater than $L$, running our guessing algorithm will always result in the suffix of length exactly $L$. Using this technique we will never deal with suffixes of length smaller than $L$.

That is why instead of choosing the guessing function for all vertices at once, we will rather divide all nodes into $levels$ for which we will choose the guessing function independently. The *level* is a set of nodes in a tree laying at the same depth. After doing the optimization for each level, the resulting function is just a combination of those partial results. Using this approach, the guessing function which is defined only at a certain tree depth $k < L$, may be applied to all words laying in the tree at a depth not smaller than $k$, for which we assume that our algorithm will always result in the node at the depth $k$. Note that when constructing a resulting function from these several partial functions, each word will be used several times for computations which will give us more statistical data.

To clarify, let us write it in a more formal way. We are iterating through levels in a tree. For the $k$-th level (that is for suffixes of length $k$):

- Let $A_k$ be a set of all nodes at the level $k$.
- Assume that we have our guessing function $S_k$ defined only for the nodes in $A_k$. We will apply it only to words in the training set of length greater than or equal $k$. If we pick up a node $a$ from $A_k$, all the words from its subtree will be assigned tags $S_k(a)$.
- Knowing what are the tags returned by the algorithm for all words of length greater than or equal $k$, we may compute the $F$-measure of obtained results describing how our guessing function behaves for these words and choose the function which maximizes this measure - let's call it $S_k^{max}$.

Our resulting guessing function $S$ is defined for all tree nodes by the equation

$$S(a) = S_k^{max}(a)$$

for every $k$ and $a \in A_k$.

### 2.4 Computing the *F-Measure*

Assume that we have a test set $T$ and a chosen guessing function $S$. We define $F$-measure for $T$ in a standard way which we will recall here for the sake of consistency. For each word $w$ in $T$ let $gold_S(w)$ denote the number of tags returned by the dictionary, $guess_S(w)$ - the number of tags returned by our algorithm, and finally $common_S(w)$ - the number of tags common for both these results. Let us define

$$Precision(S) = \frac{\sum_{w \in T} common_S(w)}{\sum_{w \in T} guess_S(w)}, \quad Recall(S) = \frac{\sum_{w \in T} common_S(w)}{\sum_{w \in T} gold_S(w)}$$

$$F(S) = \frac{2}{Precision(S)^{-1} + Recall(S)^{-1}}.$$

It's easy to observe that maximizing $F(S)$ can be transformed into maximizing the following

$$F_\alpha(S) = \frac{2\alpha \sum_{w \in T} common_S(w)}{\sum_{w \in T} gold_S(w) + \alpha \sum_{w \in T} guess_S(w)}$$

where $\alpha$ indicates how much we want to emphasize precision over recall (observe that for $\alpha = 1$ we obtain $F_1(S) = F(S)$).

### 2.5 Avoiding Exponential Growth

In this section we provide a formal and technical description of a core procedure we use to reduce the number of functions to test. From now on let us assume that we have chosen the depth at which we are defining our guessing function - the level of all suffixes of length $k$, which we will denote by $A$. The functions are tested using the set of words of length greater than or equal $k$. Let's call this set $Q$. Also, for $w \in A$, let $Q(w)$ denote the set of words from $Q$ in a subtree of $w$. Unfortunately, there is an enormous number of guessing functions to try. In order to be able to test them all we need to look deeper at our formula for computing $F$. First of all let us observe that

$$\sum_{w \in Q} common_S(w) = \sum_{w \in A} \sum_{t \in S(w)} count(w, t)$$

where $count(w, t)$ is defined as a number of words in $Q(w)$ for which $t$ is one of the tags recognized by the dictionary. Indeed, as $Q = \bigcup_{w \in A} Q(w)$, we may write

$$\sum_{w \in Q} common_S(w) = \sum_{w \in A} \sum_{u \in Q(w)} common_S(u).$$

What is more, every word in $Q(w)$ will receive the same set of tags âĂŞ $S(w)$, which means that it is enough to sum over all tags from $S(w)$ counting the number of occurrences of this tag among all words in $Q(w)$.

The second most important equality is

$$\sum_{w \in Q} guess_S(w) = \sum_{w \in A} (|Q(w)| \times |S(w)|).$$

Indeed, as we already know, every word in $Q(w)$ receives the same set of tags $S(w)$, showing that both sides of the equality above describe the total number of tags received by words in $Q$. Considering that

$$\sum_{w \in Q} gold_S(w) = C$$

as a constant not depending on our function can be computed beforehand, one may write

$$F_\alpha(S) = \frac{2\alpha \sum_{w \in A} \sum_{t \in S(w)} count(w, t)}{C + \alpha \sum_{w \in A} (|Q(w)| \times |S(w)|)}.$$

Let us investigate the behavior of $F_\alpha(S)$ if we fix $S$ for all nodes except of a certain $w$. The following lemma will make use of our newly derived formula.

**Lemma 1.** *Let $w \in A$. Assume that there are two tags $t_1 \in S(w)$, $t_2 \notin S(w)$ such that $count(w, t_2) > count(w, t_1)$. For $S^+$ being modification of $S$ satisfying $S^+(w) = (S(w) \setminus \{t_1\}) \cup \{t_2\}$ and $S^+(v) = S(v)$ for $v \neq w$, we obtain $F_\alpha(S^+) > F_\alpha(S)$. In particular, $S$ is not optimal.*

*Proof.* The result is fairly straightforward, as when modifying $S$ to obtain $S^+$, we note that in our derived formula for $F_\alpha(S)$ the value of denominator does not change as $|S(v)| = |S^+(v)|$ for all $v$, while the value of the nominator increases, given our choice of elements to perform a swap in $S(w)$ .

The lemma shows that in order to obtain $S$ maximizing $F_\alpha(S)$, for each $w$ in $A$ we need to sort the list of all tags associated with the node $w$ (our *morphological information* assigned to $w$) in a decreasing order regarding $count(w, t)$ and choose first $x(w)$ ($x$ is any function $x : A \to \mathbb{N}$) of them. Otherwise $S$ would not be optimal as we could achieve a better result transforming it to $S^+$ using the lemma above. Using this observation, we have in fact significantly reduced the number of strategies to consider. This amount is still exponential, but much easier to tackle with our second step of optimization. Note that now we are facing the task of finding the maximum of the $|A|$ dimensional discrete function (indeed, function $x$ can be considered as a point in a $|A|$ dimensional space). We cannot directly compute all the values. Instead, we need some cost-effective, incremental iterative techniques. Beginning with a starting point

- Iterate through words $w$ in $A$, for each word finding the value $x(w)$ maximizing $F_\alpha$, given that values of $x$ for all other words remain fixed. In other words we find a maximum, treating $F_\alpha$ as a function in one variable $x(w)$. Set $x(w)$ to the newly computed value.
- Iterate this process as long as we managed to increase the value $F_\alpha$ during the last step.

It can be observed that this iterative approach has to reach the end, as during each step we move to the point augmenting $F_\alpha$, while the number of different points is finite. During extensive tests we have noted that this process always finishes after at most 4 iterations. It is an open question whether the final point is indeed the global maximum (we know that it is the local maximum). Our implementation, performing such an incrementation starting from 20 randomly chosen points, always reaches the same ending point, which makes us believe it is indeed the global maximum.

## 3 Further Optimisations

Approach discussed so far is applicable to guessers of any inflectional language. The rest of this section will be specific to the Polish language. There are several features of Polish which can be determined based on the word's prefix rather than suffix. This includes 3 features

- *Nie*, being the prefix of the word helps in determining whether it is negated or not.
- *Naj* at the beginning of an adjective often indicates the superlative form.
- Prefix of the verb has an impact on the perfection.

These optimizations can be used to modify the resulting set of tags before returning it to the user.

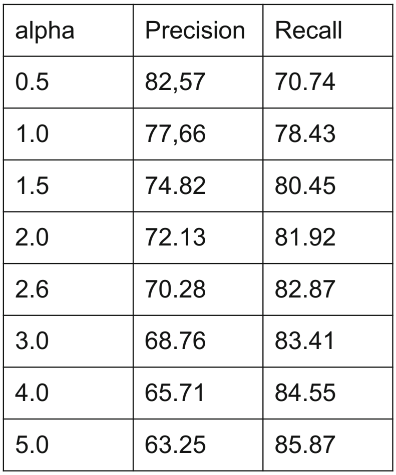
## 4 Testing and Results

Theoretical approach described in this paper has been implemented as a proof-of-concept. Preliminary results are included here for the sake of completeness.

Tests were performed using the ten-fold cross validation schema. Manually annotated part of the National Corpus of Polish ([2]) was used as the reference set of words.

From the set of words known to the dictionary, we have chosen those constisting of more than 3 letters. The resulting set contained roughly 120000 words. The set has been divided randomly into ten equal parts and the following schema was applied 10 times.

We pretend that one of the parts is not known to the dictionary - this is our testing part. The rest is all the knowledge the dictionary has - this is the training part. Now, we compute the results using formulae defined in section 2.4. The following data was obtained with different values of $\alpha$



| alpha | Precision | Recall |
|---|---|---|
| 0.5 | 82,57 | 70.74 |
| 1.0 | 77,66 | 78.43 |
| 1.5 | 74.82 | 80.45 |
| 2.0 | 72.13 | 81.92 |
| 2.6 | 70.28 | 82.87 |
| 3.0 | 68.76 | 83.41 |
| 4.0 | 65.71 | 84.55 |
| 5.0 | 63.25 | 85.87 |

**Fig. 2.** Preliminary results of the guesser implementation

## 5 Evaluation

The approach introduced in this paper produces relatively good results. Comparing to the results from Piasecki and Radziszewski ([1]) achieving precision 70.3 and recall 71.7 we get 1.1 pp. (percentage point) improvement on the recall. We cannot compare these approaches in a better manner because of different kinds of training sets. We may observe that increasing recall causes precision to decrease which is understandable.

Please note that the main purpose of these tests was to show that our theoretical approach can be succesfullly used in practice. Practical implementations making use of specific training sets are welcome. One may think about improvements on the presented approach. We may try to allow a guessing function for some *communication* between levels, so that we would maximize the $F$-measure globally. One can also treat this approach as a baseline, building on the top of it the set of sophisticated guessing rules.

Our approach itself (trying to apply every possible guesser to the training data) is interesting from a methodological point of view. Presented technique can be easily generalized and the initial proof-of-concept implementation produces satisfying results. The idea of finding the optimal strategy by reducing significantly the problem size is also an interesting example of applying algorithmical and mathematical techniques to optimization problems.

## References

1. Piasecki, M., Radziszewski, A.: Morphological prediction for Polish by a statistical a tergo index. Systems Science 34, 7–17 (2008)
2. National Corpus of Polish, `http://nkjp.pl`

# Prefix Recognition Experiments[*]

Jaroslava Hlaváčová and Michal Hrušecký

Charles University in Prague, ÚFAL MFF
hlavacova@ufal.mff.cuni.cz, michal@hrusecky.net

**Abstract.** The paper deals with automatic methods for prefix extraction and their comparison. We present experiments with Czech and English and compare the results with regard to the size and type (wordforms vs. lemmas) of input data.

## 1 Introduction

Prefixation and/or suffixation is one of the major means of word-formation in many languages. Prefixes and suffixes serve also for word inflection in flective languages.

Sets of affixes are usually well known for a given language, together with their main functions or meanings. The same thing may be stated about prefixed and suffixed words — they are usually included in dictionaries and there is no need to invent them again and again.

However, as languages evolve, new affixes may appear to create new words. One of the main sources of the new affixes are foreign languages.

Let us take as an example the prefix *re* closely connected with repeating events. There are quite a lot of words with this prefix in Czech corpora. We present several examples from the corpus SYN2010 proving that the prefix *re* may be attached to nouns, adjectives and verbs having always the meaning of an action, event. The prefix *re* adds the meaning of repetition.

*renormalizace, renormalizovatelnost, renormalizovat*
*redesign, redesignovat, redesignovaný*
*remodelace, remodelování, remodelovat*

On the other hand, the language itself often serves as a repository of new affixes — they are usually transformed roots. Taken strictly, they are not prefixes in the very linguistic sense. Pure linguists would probably call them rather stems and the resulting words compounds. However, they behave like prefixes — they can be attached to many existing words, changing their meaning, very often in the same or similar way.

Typical examples are names of colours usually attached to adjectives and adverbs, not so often to nouns. The following examples were taken again from the corpus SYN2010. We extracted some of those that were not present in the morphological dictionary:

[*] This paper is a result of the projects supported by the grants number P406/2010/0875 and P202/10/1333 of the Grant Agency of the Czech republic (GAČR), and the grant MSM 0021620838 of the Czech Ministry of Education.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 235–242, 2011.

*žlutoběžová, žlutozelená, žlutorukých, žlutodřevu, žlutohněda,*
*hnědoočky, hnědopurpurovými, hnědoběžovou, hnědočervena, hnědopyský*

For an automatic language processing it is very convenient to have a list of all affixes for a given language, as it helps to recognize "new"words that were not included into dictionaries (due to various reasons). The recognition consists not only of guessing morphological properties of the "new"words, but usually also their meanings. Thus, it is possible to use such lists for synthesis too.

At the first glance one could expect that for a given language, there exists a complete list of all its affixes, or at least that such a list is easy to collect. It is not the case however.

There are several statistical methods how to extract affixes automatically from a large amount of words. Their overview can be found for instance in [1]. We implemented some of them into a complex tool Affisix [2] and used it for experiments with several languages. Some of them are described in [3], [4] and [5]. In this contribution, we present selected results of prefix extraction, processed on three big Czech corpora, namely SYN2000, SYN2005 and SYN2010, and British National Corpus, each having 100 million tokens. We concentrate especially at differences among the methods and among the properties of the intput data.

## 2 Methods

In this section, we briefly introduce methods we used for our recent experiments. The detailed description may be found in [1]. All the methods need as an input a long list of words or lemmas. Every word is divided into two or more strings — segments — and investigated, if the segmentation has certain properties or not. The properties are expressed numerically, so it is easy to compare different segmentations and select those that are the best (for instance the highest). Segments that pass certain threshold are marked and we call them prefix-candidates.

### 2.1 Naive Method

This method is based on two simple assumptions, but it produces quite interesting results. The first assumption: prefixes can be attached to many words. The second assumption: if a string is a prefix, we can remove it and the rest is still a meaningful word. The second assumption is not true for many prefixes, but for searching the "new"prefixes in the language, it works very well. These prefixes are usually simply glued to the beginning of existing words.

For every initial segment we count number of words starting with that segment and the number of words this segment can be attached to as a prefix.

$$n_p = |\{x; x \in S \;\&\; \exists y; x = p :: y\}| + |\{x; x \in S \;\&\; p :: x \in S\}|$$

where S is a set of all meaningful words and :: denotes concatenation. The greater the number $n_p$, the greater chance that the segment $p$ is a real prefix.

### 2.2 Squares

A square is a quadruple $< a, b, c, d >$ of strings such that any combination of the first two strings with the second two strings forms a valid word in the language. Any of the strings can be empty. In this method, we look at every initial segment and count the number of squares it is in. Bigger the number of squares the segment is in, more probable the segment is an affix. In contrast to the previous naive method, the Squares method recognizes even prefixes in words where prefix is obligatory (for instance in the *jednoruký*).

### 2.3 Entropy Methods

This method is based on the observation that the entropy between morphemes is usually higher than elsewhere. After a prefix, the entropy increases because the prefix string may be followed by many other strings, which is not the case inside the prefix, nor inside a subsequent stem. Thus, we can check the entropies after initial strings of input words, sort them and take those with the highest values as good candidates for prefixes.

Entropy is in general computed using the following formula:

$$H(a) = -\sum_{s \in C} p(s|a) \log_2 p(s|a)$$

where $C$ is a set of possible continuations of the beginning string $a$.

We modified this approach by taking a difference between entropies of two adjacent letters instead of the entropy itself. We call this modification the difference entropy.

The list of prefix candidates according to the difference entropy gives usually better results, which means that among first $N$ prefix candidates with the highest values of difference entropy we can find more real prefixes than among the same amount of candidates extracted by the (pure) entropy method.

See the results in the section 4.

### 2.4 Economy Method

For the description of this method, we cite from [1]: "If a word is divided in two segments, one of them occurring in many other words, while the other occurs in only a few others, and if the first one belongs to a small set of frequent segments, while the other to a potentially infinite set of low occurring segments, then a morphological cut can be proposed between these segments."

Thus, the set of possible prefixes is obtained as a list of segments from the beginning of the words that have more possible continuations than the rest of the word has possible beginnings.

For a segmentation of a word, the economy index is calculated as a ratio of sizes of two sets: size of subset of all possible continuations of the initial segment divided by a size of subset of possible beginnings of the ending segment. For a detailed description of the subsets and the method itself see [1].

## 3 Data

We were mainly interested in method results using different sets of data we performed experiments with several different corpora. We used SYN2000, SYN2005 and SYN2010 [6] corpora for the Czech language and BNC [7] for English. For all the corpora, we extracted lists of wordforms with the frequency more than 10 and 50, respectively. For Czech, we made similar lists for lemmas as well. These sets of data were selected in order to test how the amount of data would influence the results of individual methods and whether it is better to use wordforms or lemmas for the prefix recognition. For English we used only word forms and we were mainly interested whether results of methods comparison in Czech would reflect in English as well.

Table 1 shows number of tokens (words or lemmas) entering the experiments with different corpora. Lists of tokens that we used for our experiments are named after their properties visible from the table 1: for instance syn2005-word-10 is the name of the list of words from the corpus SYN2005 with the frequency more than 10.

**Table 1.** Corpora comparison

| **corpus** | **word-50** | **word-10** | **lemma-50** | **lemma-10** |
|---|---|---|---|---|
| syn2000 | 114 283 | 305 677 | 56 302 | 123 018 |
| syn2005 | 118 838 | 319 156 | 56 639 | 122 831 |
| syn2010 | 116 266 | 311 143 | 54 836 | 117 961 |
| bnc | 48 074 | | | |

## 4 Experiments

For each method, we sorted the numerical characteristics of the prefix-candidates. Then, we manually checked the best 100 and selected those that act as real prefixes.

Each method was assigned the score acquired by subtracting number of bad results from 100 (number of all results). Naturally, the score is a function of number of prefix-candidates — see examples of graphs in the following subsections.

We also present a table 2 showing several prefix candidates for entropy methods. For other methods, we only briefly describe their results.

**Naive Method.** Though this method is very simple and its assumptions may not be always fulfilled, it gave quite good results. The results do not differ much for individual corpora. In other words, the score defined above decreases for all the corpora roughly equally.

**Squares Method.** Surprisingly, this method performs considerably worse than naive method. It is also the slowest one of all the tested methods. Again, there is no considerable difference among the corpora.

**Economy Principle.** This method does not perform very well, but there is a small difference in favor of lemmatized corpora as compared with their un-lemmatized counterparts.

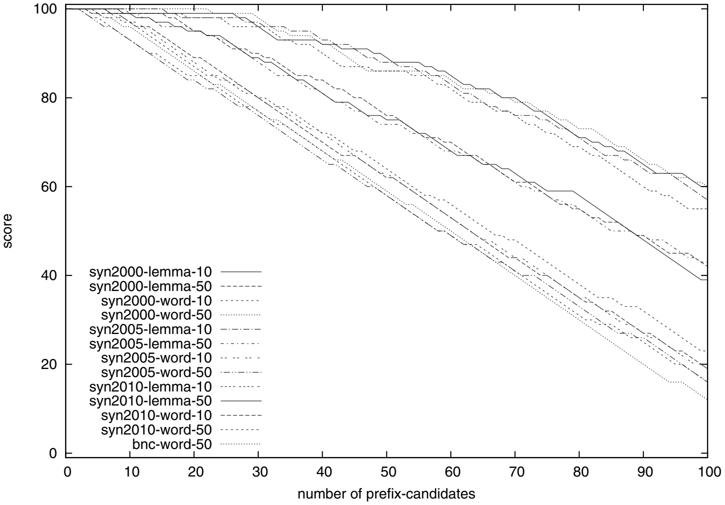


**Fig. 1.** Comparison of results of difference entropy method

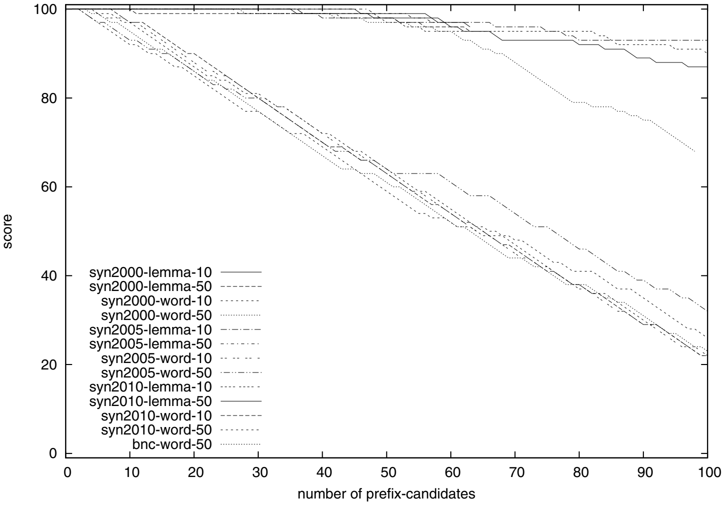


**Fig. 2.** Comparison of results of filtered difference entropy method

**Entropy Methods.** For this method, we present two graphs showing the difference among the input data.

The results of difference entropy method are presented in the graph 1. There are three main clusters of lines. The lowest one, showing the worst results, was obtained using Czech wordforms. The reason is that the difference entropy tends to prefer longer prefixes. As cuts between a stem and a suffix are more obvious than between a prefix and a root, the difference entropy method got distracted many times by the suffixes.

On the contrary, English wordforms scored as good as the best Czech lemmas even though the English list is the smallest one. It can be explained by the not so rich English morphology — there are not many suffixes to distract the method.

The two upper clusters of the graph show, that amount of words matters. Both are results of lemmas, but the higher cluster belongs to the lists filtered with the frequency 10, while the middle with frequencies more than 50.

We tried additional filtering — using a constraint similar to the second assumption used for the naive method (see sec. 2.1). We demanded that among the words $x = p :: w$ with a prefix $p$, there must be at least 10 words, for which $w$ is also a word. Unfortunately it turned out (see graph 2) that this constraint isn't limiting enough. Although it improved results, especially for smaller lists (these are now in the same cluster as their bigger counterparts), it still didn't improve the performance on wordforms. Neither did it improve results of BNC much (worst line from the upper cluster).

**Table 2.** Top ten prefix-candidates from entropy methods (prefixes are bold)

| Filtered difference entropy | | | Entropy | | |
|---|---|---|---|---|---|
| rank | score | prefix | rank | score | prefix |
| 1. | 2.5996971 | **over** | 1. | 2.8503499 | **non-** |
| 2. | 2.4349861 | **micro** | 2. | 2.7454889 | **inter** |
| 3. | 2.4228690 | **water** | 3. | 2.6891198 | *mar* |
| 4. | 2.4150519 | **school** | 4. | 2.6787214 | **back** |
| 5. | 2.3911490 | **black** | 5. | 2.6780901 | **pro** |
| 6. | 2.3825233 | **super** | 6. | 2.6724777 | **over** |
| 7. | 2.2052698 | **stock** | 7. | 2.6367834 | *car* |
| 8. | 2.0895109 | **light** | 8. | 2.6299610 | **under** |
| 9. | 2.0889339 | **under** | 9. | 2.6107635 | *cra* |
| 10. | 2.0280159 | **self-** | 10. | 2.5970426 | *man* |

### 4.1 Comparison of All Methods

The last experiment compares all the methods on the same list. Here we present only the graph for the results from the list syn2010-lemma-10 ( Figure 3).

The experiments conducted on the other lists gave very similar results, the only visible difference being between lists of lemmas and wordforms. The former were always more successful, so it is better to use lemmatized data rather than wordforms. If a lemmatized corpus is not available, we recommend to use the naive approach to limit the search to prefixes only. The squares method and economy principle didn't perform well in our tests. On the other hand, entropy, and especially difference entropy performed well and were fast to compute. Surprisingly naive approach performed much better than more complicated methods.

All these facts can be derived from table 3 with the overall results from all the experiments.

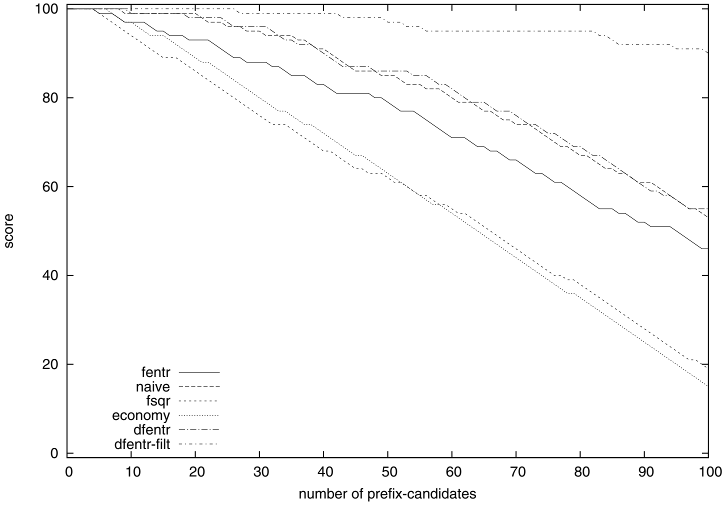


**Fig. 3.** Comparison of results on the list syn2010-lemma-10

**Table 3.** Comparison of precision of all methods for top 50 prefix-candidates

| | naive | squares | economy | dentr | dentr-filt |
|---|---|---|---|---|---|
| **syn2000-lemma-10** | 66 % | 20 % | 26 % | 78 % | 98 % |
| **syn2000-lemma-50** | 66 % | 18 % | 24 % | 52 % | 96 % |
| **syn2000-word-10** | 82 % | 26 % | 38 % | 28 % | 28 % |
| **syn2000-word-50** | 66 % | 16 % | 34 % | 18 % | 22 % |
| **syn2005-lemma-10** | 74 % | 22 % | 28 % | 76 % | 96 % |
| **syn2005-lemma-50** | 66 % | 20 % | 24 % | 48 % | 98 % |
| **syn2005-word-10** | 80 % | 28 % | 38 % | 26 % | 26 % |
| **syn2005-word-50** | 64 % | 16 % | 36 % | 16 % | 28 % |
| **syn2010-lemma-10** | 70 % | 24 % | 26 % | 72 % | 94 % |
| **syn2010-lemma-50** | 70 % | 22 % | 24 % | 50 % | 96 % |
| **syn2010-word-10** | 78 % | 22 % | 40 % | 24 % | 26 % |
| **syn2010-word-50** | 68 % | 16 % | 36 % | 16 % | 18 % |
| **bnc-word-50** | 66 % | 24 % | 18 % | 72 % | 94 % |

## 5 Conclusions and Plans

Experiments conducted so far suggest that results of individual methods depend on the size of the input data (the corpus and its filtration) and language. For practical use (e.g. guessers of OOV[1] words or building morphematic databases) it would be important to select the appropriate method and the input corpus. Whenever it is possible, it is better to use lemmatized data rather than wordforms.

[1] Out of vocabulary.

In the future we plan to continue with experiments and try to improve the results especially by some additional constraints or using combinations of the methods.

We also plan to try using these method for unsupervised stemming and compare the results against those of basic language-specific stemmers.

## References

1. Urrea, A.M.: Automatic discovery of affixes by means of a corpus: A catalog of spanish affixes. Journal of Quantitative Linguistics 7, 97–114 (2000)
2. Hrušecký, M.: Affisix, http://affisix.sf.net
3. Hrušecký, M., Hlaváčová, J.: Automatické rozpoznávání předpon a přípon s pomocí nástroje affisix. In: Pardubská, D. (ed.) Informačné technológie Aplikácie a Teória, Zborník príspevkov prezentovaných na konferencii ITAT, Seňa, Slovakia, PONT s. r. o, pp. 63–67 (2010)
4. Bojar, O., Straňák, P., Zeman, D., Jain, G., Hrušecký, M., Richter, M., Hajič, J.: English-hindi translation obtaining mediocre results with bad data and fancy models. In: Sharma, D., Varma, V., Sangal, R. (eds.) Proceedings of ICON 2009: 7th International Conference on Natural Language Processing, Hyderabad, India, NLP Association of India, pp. 316–321. Macmillan Publishers, India (2009)
5. Hlaváčová, J., Hrušecký, M.: "affisix" tool for prefix recognition. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 85–92. Springer, Heidelberg (2008)
6. Ústav Českého národního korpusu FF UK: Český národní korpus - syn2000, syn2005, syn2010 (2000), http://ucnk.ff.cuni.cz
7. Oxford University Computing Services on behalf of the BNC Consortium: The british national corpus (2007), http://www.natcorp.ox.ac.uk

# Question Classification by Weighted Combination of Lexical, Syntactic and Semantic Features

Babak Loni, Gijs van Tulder, Pascal Wiggers, David M.J. Tax, and Marco Loog

Delft University of Technology, Pattern Recognition Laboratory,
P.O. Box 5031, 2600 GA Delft, The Netherlands
{b.loni,G.vanTulder}@student.tudelft.nl,
{P.Wiggers,D.M.J.Tax,M.Loog}@tudelft.nl

**Abstract.** We developed a learning-based question classifier for question answering systems. A question classifier tries to predict the entity type of the possible answers to a given question written in natural language. We extracted several lexical, syntactic and semantic features and examined their usefulness for question classification. Furthermore we developed a weighting approach to combine features based on their importance. Our result on the well-known TREC questions dataset is competitive with the state-of-the-art on this task.

## 1 Introduction

One of the most crucial tasks in Question Answering (QA) systems is question classification. A question classifier predicts the *entity type* of a possible (factual) answer for a given question. For example, if the system is asked "What is the capital of the Netherlands?", the question classifier should assign to this question the label *city*, since the expected answer is a named entity of type *city*.

Determining the class of a question is quite useful for the process of answering the question. Knowing that the question is of a particular type, the search space for possible answers can be narrowed down to a much smaller space. Furthermore, the question class can be used to rank the candidate answers [5,12].

In this work, we developed a feature-driven learning-based question classifier that is competitive with state-of-the-art question classification approaches. We extracted known and new lexical, syntactic and semantic features and compared the classification accuracies that can be obtained with these sets. Furthermore, we investigated whether combining feature sets can improve classification accuracy. For this we introduce a weighted combination approach that takes into account the importance of the features.

This paper is organized as follows. We start with a discussion of related work in section 2. In section 3 we discuss our motivation for choosing support vector machines (SVMs) as our classifier. In section 4 we explain the features that we extracted and their individual classification accuracies. We introduce our approach to combine features in section 5. We end with a conclusion.

## 2 Related Work

Different approaches to question classification have been proposed. Some early studies build question classifiers based on matching with hand-crafted rules [15]. However,

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 243–250, 2011.

these approaches do not generalize well to new domains and do not scale easily. Most recent studies build question classifiers based on machine learning approaches [6,16,9,10] that use features extracted from the question.

The accuracy of most of the studies, including this work, is usually measured on a well-known *taxonomy* of question classes proposed by Li and Roth [9]. This taxonomy has two layers consisting of 6 coarse grained and 50 fine grained classes (Table 1). A dataset of almost 6000 labeled question has been created based on this taxonomy[1] [9]. This dataset which is usually referred to as the TREC (Text REtrieval Conference) dataset, is divided in a training set of 5500 questions and a test set of 500 questions. The accuracy of a question classifier is defined as the number of correctly classified questions divided by total number of questions.

**Table 1.** The coarse and fine grained question classes

| Coarse | Fine |
|---|---|
| ABBR | abbreviation, expansion |
| DESC | definition, description, manner, reason |
| ENTY | animal, body, color, creation, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word |
| HUM | description, group, individual, title |
| LOC | city, country, mountain, other, state |
| NUM | code, count, date, distance, money, order, other, percent, percent, period, speed, temperature, size, weight |

Li and Roth [9] obtained an accuracy of 84.0% on the fine grained classes using a SNoW (Sparse Network of Winnows) architecture. Using different semantic features, [10] obtained an accuracy of 89.3% for the fine grained classed. [6] reached accuracies of 89.0% using a Maximum Entropy model and 89.2% using SVMs with linear kernels on the fined grained classes, while they obtained an accuracy of 93.6% on the coarse grained classes. Zhang et. al. [16] proposed a syntactic tree kernel for SVM-based question classification. They obtained an accuracy of 90.0% on the coarse grained classes. Pen. et. al. [11] reach an accuracy of 94.0% on the course grained classes with a semantic tree kernel for SVM classifiers. [13] combined rule-based and learning based approaches. They used matched rules as features for a SVM classifier. They reach an accuracy of 90.0% on the fine grained and an accuracy of 94.2% on the coarse grained classes. To our knowledge this is the highest accuracy achieved on the TREC dataset.

## 3 Choosing the Classifier

The choice of classifier is an important decision in our system. Since in the question classification problem the questions are represented in a very high dimensional feature space, we decided to choose Support Vector Machines [14] as our classifier. SVMs are

[1] http://cogcomp.cs.illinois.edu/Data/QA/QC/

shown to have good performance on high dimensional data and generally outperform other classifiers, such as Nearest Neighbor, Naive Bayes, Decision Trees, SNoW and Maximum Entropy on question classification [16,6,5]. We decided to rely on simple linear kernels for the SVMs together with rich features, rather than on other, more complex kernels. All systems were implemented with LIBSVM [1], a library for support vector machines.

## 4 Features in Question Classification

We used three different types of features: lexical, syntactic and semantic features. We introduce each type of feature and show classification results than can be achieved for every feature type.

### 4.1 Lexical Features

Lexical features of a question are generally based on the *context words* of the question, i.e., the words that appear in a question. In the *unigram* or *bag-of-words* approach each word in the vocabulary is treated as a feature. For each question the value of every word feature is set to the frequency count of that word in the question. This can lead to a high dimensional feature space, but that can be dealt with by using sparse representations that only store non-zero entries. Unigram features are a special case of $n$-gram features, that treat every sequence of $n$ consecutive words in the question as a feature.

To obtain insight in the influence of lexical features on question classification, we trained our classifier with different types of lexical features. The classification accuracy is listed in Table 2. It shows that, most likely due to data sparseness, unigrams are better features than bigrams.

In recent studies Huang et. al [5,6] considered question *wh-words* as a separate feature. They selected 8 types of wh-words, namely *what*, *which*, *when*, *where*, *who*, *how*, *why* and *rest*. For example the wh-word feature of the question "What is the longest river in the world?" is *what*. We extracted wh-word features from a question with the same approach as [5].

The *word shape* is a word-level feature that refers to the type of characters used in a word. This can be useful to identify for example numerical values and names. Inspired by [6], we introduced four categories for word shapes: *all digit*, *lower case*, *upper case* and *other*. Not surprisingly, Word shapes alone is not a good feature set for question classification (Table 2), but, as will be shown in the next section, combined with other features they can improve classification accuracy.

**Table 2.** The accuracy of SVM classifier based on different lexical features for coarse and fine grained classes

| **Feature Space** | unigram | bigram | wh-words | word-shapes |
|---|---|---|---|---|
| **Coarse** | 88.2 | 86.8 | 45.6 | 35.5 |
| **Fine** | 80.4 | 75.2 | 46.8 | 30.8 |

### 4.2 Syntactic Features

A different class of features can be extracted from the syntactic structure of the question. We extracted two syntactical features namely *Tagged Unigrams* and *Head Words*.

**Tagged Unigrams:** We introduce a new feature namely *tagged unigram* which are unigrams augmented with their part-of-speech (POS) tags. They allow the model to differentiate between words that have the same lexical form (e.g. the verb 'to record' vs. the noun 'record'). We used the Stanford implementation of a Maximum Entropy POS-tagger [7] to tag the questions. Following is a sample question from the TREC dataset augmented with its POS-tags:

Who_WP was_VBD The_DT Pride_NNP of_IN the_DT Yankees_NNPS ?_

Similar to the bag-of-words approach, the bags are now made up of augmented words. Most likely due to data sparseness, tagged unigrams do not necessarily have a better accuracy than unigrams (Table 3). but when combined with other features, they sometimes show better performance compared to unigrams.

**Head Words:** For question classification the head word is usually defined as the "single word that specifies the object that the question seeks" [6]. For example for the question "What is the oldest city in the United States ?", the head word is "city". The head word is usually the most informative word in the question and correctly identifying it can significantly improve the classification accuracy.

Extracting the head word of a question is quite a challenging problem. Similar to [6,13], we extracted the head word based on the syntactical structure of the question. To obtain the head word, we first parse the question using the the Stanford parser [7] and then extract the head word based on the parse found. For head word propagation we adapted the rules defined by [2] — for the propagation of syntactic heads — to prefer noun heads over verb heads, as for question answering the subjects and objects of a sentence are usually more informative than its verbs.

Table 3 lists the accuracy of two syntactic features that we used in this work.

**Table 3.** The accuracy of SVM classifier based on different syntactic features for coarse and fine grained classes

| **Feature Space** | tagged unigram | Headwords |
|---|---|---|
| **Coarse** | 87.4 | 62.2 |
| **Fine** | 80.6 | 40.6 |

### 4.3 Semantic Features

In addition to lexical and syntactic features, we extracted two features related to the semantics of the question called *related words group* and *Hypernyms*.

**Related Words Group:** Li and Roth [10] defined groups of words, each represented by a category name. If a word in the question exists in one or more groups, its corresponding categories will be added to the feature vector. For example if any of the words {birthday, birthdate, day, decade, hour, week, month, year} exists in a question, then its category name *date* will be added to the feature vector.

**Hypernyms:** For a given word, a hypernym is a word with a more general meaning. For example a hypernym of the word "city" is "municipality". As hypernyms allow one to abstract over specific words, they may be useful features for question classification [5]. We used WordNet [3] together with the MIT Java Wordnet Interface package [4] to extract hypernyms.

However, extracting hypernyms is not straightforward. There are four challenges that should be addressed to obtain hypernym features: 1) For which word(s) in the question should we find hypernyms? 2) For the candidate word(s), which part-of-speech should be considered? 3) The candidate word(s) augmented with their part-of-speech may have different senses in WordNet. Which sense is the sense that is used in the given question? and 4) How far should we go up through the hypernym hierarchy to obtain the optimal set of hypernyms?

To address the first issue, we choose the head word as the candidate word, since its the most informative word in a question. We found that considering (also) other words in a question can introduce noisy information in feature vector and leads to lower accuracy.

For the second issue we used the POS tags extracted for the syntactic features. To tackle the third issue we adopted Lesk's Word Sense Disambiguation (WSD) algorithm to find the right sense of the candidate word. Lesk's WSD algorithm [8] is a dictionary-based method for resolving the true sense of a word in a sentence. It looks at the descriptions of different senses of the candidate word and chooses the sense in which the description has maximum similarity with the description of the context words in the sentence.

Finally, to address the 4th challenge, i.e. the depth of hypernyms to use, we relied on the experiments of [5], in which the value of six is considered as the maximum depth of hypernyms. Table 4 lists the accuracies of the semantic features for question classification. The interesting point about the results in Table 4 is that the "Related words group" features alone have better performance than lexical features. This shows the importance of semantic features in question classification. Hypernyms did not result in a good classification accuracy. The reason may lie in the complicated sequence of tasks needed to extract the hypernyms; an incorrect decision in any task can increase the noise in the feature vector.

**Table 4.** The accuracy of SVM classifier based on different semantic features for coarse and fine grained classes

| **Feature Space** | related word groups | Head hypernyms |
|---|---|---|
| **Coarse** | 85.2 | 66.6 |
| **Fine** | 79.8 | 41.6 |

## 5 Combining Features

The three feature sets we described each take a different perspective on the question. We explored whether combining different feature sets will improve the classification accuracy. Unlike related work in which the augmented features are blindly added to the feature vector, we suggest a weighted concatenation of the various feature sets:

$$f = (w_1 f_1^{\mathrm{T}}, \ldots, w_m f_m^{\mathrm{T}})^{\mathrm{T}} \tag{1}$$

where $f_i$ is the $i^{th}$ feature set, $w_i$ is its weight, $m$ is the number of feature sets that are extracted and $f$ is the final feature set. In total we implemented 9 types of different features, i.e, $m = 9$. If $w_i = 0$ it means that the $i^{th}$ feature set will not be added to the final feature set. Table 5 lists the classification accuracies based on different combinations of features with equal weights (1.0) on the standard TREC test set.

**Table 5.** The accuracy of SVM classifier based on different combinations of feature sets, with equal weights, on coarse and fine grained classes

| No. | Feature Set | Coarse | Fine |
|---|---|---|---|
| 1 | unigram | 88.2 | 80.4 |
| 2 | unigram + wh-words | 88.2 | 80.4 |
| 3 | unigram + head words | 89.0 | 84.0 |
| 4 | unigram + hypernyms | 90.2 | 84.6 |
| 5 | unigram + related words group | 90.0 | 85.2 |
| 6 | unigram + related words group + word shapes | 89.8 | 86.2 |
| 7 | (6) + tagged unigram | 90.6 | 86.2 |
| 8 | (6) + bigram | 92.0 | 86.6 |
| 9 | (6) + head words | 90.8 | 86.4 |
| 10 | (6) + head words + hypernyms | 91.4 | 88.0 |
| 11 | (6) + head words + hypernyms + bigram | 93.2 | 88.0 |

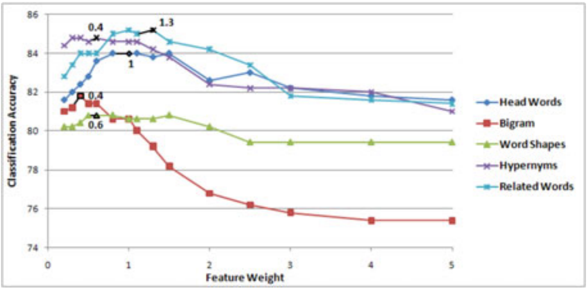


**Fig. 1.** Classification accuracy of unigram features combined with an other feature set as a function of the combination weight

**Table 6.** Confusion matrix showing the classifications of the TREC-500 for the coarse categories

| | | **Predicted labels** | | | | | |
|---|---|---|---|---|---|---|---|
| | | ABBR:* | DESC:* | ENTY:* | HUM:* | LOC:* | NUM:* |
| **True** | ABBR:* | 9 | | | | | |
| | DESC:* | | 134 | 2 | | 1 | 1 |
| | ENTY:* | | 10 | 83 | 1 | | |
| | HUM:* | | 1 | 1 | 63 | | |
| | LOC:* | | 1 | 9 | | 71 | |
| | NUM:* | | 3 | 2 | | | 108 |

The best classification accuracy is obtained with the combination of the following six feature sets: unigram, related words group, word shapes, head words, hypernyms and bigrams. To optimize the weight values in equation 1, we would need an exhaustive search of all possible weight assignments. As this is time-consuming, we chose a greedy approach instead. For each feature set we searched for the optimal weight when it was combined with the unigram features only. Figure 1 illustrates the classification accuracy of different features as a function of their weight. The best weight values, which are specified by a label in Figure 1, are used as weight values when combining all feature sets. This resulted in an accuracy of 89.0% on the fine grained classes and 93.6% on the coarse grained classes. We found that some questions are easier to classify than others. While the system performed well for the categories ABBR, DESC, HUM and NUM it made many more errors for the ENTY and LOC categories (Table 6).

## 6 Conclusion

We developed a learning-based, feature driven question classifier which reaches an accuracy of 89.0% on the fine grained and 93.6% on the coarse grained classes of the TREC dataset, by weighted combination of different features. We succeeded to improve the classification accuracy by almost 9% by adding different features to the basic unigram (bag-of-word) features.

We introduced the concept of a weighted combination of features on question data. Adopting the weights is an important issue when the features are combined. We could further improve the accuracy of classifier by approximating the weights to their optimal combination.

## References

1. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001) Software, http://www.csie.ntu.edu.tw/~cjlin/libsvm
2. Collins, M.: Head-Driven Statistical Models for natural Language Parsing. PhD thesis, University of Pennsylvania (1999)
3. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
4. Finlayson, M.A.: MIT Java WordNet Interface series 2 (2008)

5. Huang, Z., Thint, M., Celikyilmaz, A.: Investigation of question classifier in question answering. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), pp. 543–550 (2009)
6. Huang, Z., Thint, M., Qin, Z.: Question classification using head words and their hypernyms. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 927–936 (2008)
7. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceeding of the 41st Annual Meeting for Computational Linguistics (2003)
8. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24–26 (1986)
9. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th International Conference on Computational linguistics, pp. 1–7. Association for Computational Linguistics (2002)
10. Li, X., Roth, D.: Learning question classifiers: The role of semantic information. In: Proc. International Conference on Computational Linguistics (COLING), pp. 556–562 (2004)
11. Pan, Y., Tang, Y., Lin, L., Luo, Y.: Question classification with semantic tree kernel. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, pp. 837–838. ACM, New York (2008)
12. Quarteroni, S., Manandhar, S.: Designing an interactive open-domain question answering system. Nat. Lang. Eng. 15, 73–95 (2009)
13. Silva, J., Coheur, L., Mendes, A., Wichert, A.: From symbolic to sub-symbolic information in question classification. Artificial Intelligence Review 35(2), 137–154 (2011)
14. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc., New York (1995)
15. Voorhees, E.M.: Overview of the trec 2001 question answering track. In: Proceedings of the Tenth Text REtrieval Conference (TREC), pp. 42–51 (2001)
16. Zhang, D., Lee, W.S.: Question classification using support vector machines. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 26–32. ACM, New York (2003)

# Recursive Decompounding in Afrikaans

Tilla Fick and Chris Swanepoel

Department of Decision Sciences, University of South Africa,
Muckleneuk Campus, Preller Street, Muckleneuk, Pretoria, South Africa
{fickm,swanecj}@unisa.ac.za
http://www.unisa.ac.za

**Abstract.** An algorithm has been developed to decompose compound words in Afrikaans. This data driven technique recursively uses an extensive list of Afrikaans words in the decompounding process. String fitting from the beginning and end of words forms the basis of the process, while sublists containing short words that may occur only at the beginning or end of words, and lists of prefixes and suffixes are utilised. Applying the algorithm to the original lexicon of 182 433 words resulted in accuracy of 90,2%, precision of 99,9% and recall of 83,6%.

## 1 Introduction

Afrikaans, like Dutch and German, is a compounding language where compounds are continuously created by joining two or more existing words. For example, the word *liggaamshitte* (body heat) is compounded from *liggaam* (body) and *hitte* (heat), using the linking morpheme *s*.

For linguistic processes such as machine translation (MT) [7,12,6], speech recognition [1] and information retrieval [2] it is often necessary to determine the boundaries between the constituent parts of compound words. For example, Popović, Stein and Ney [12] found that linguistic and corpus-based compound splitting and enhanced word alignment improved German-English MT.

The eventual goal of the research on which this paper is based, is to develop a technique for automatic hyphenation in Afrikaans. The reliability of existing techniques is questionable and hyphenation needs to be checked by hand before publication. The compounding nature of Afrikaans often causes long words, increasing the need for hyphenation – especially when text is printed in narrow columns as found in magazines, journals and newspapers.

Normally, Afrikaans words are hyphenated according to syllables, but to enhance readability compounds should preferably be split at the boundaries between their constituent parts. Furthermore, it was found that hyphenation errors often occur at the boundaries between the constituent parts [5].

Existing techniques for compound splitting are mainly based on frequency, probability and string-fitting – all corpus-based methods in some way. A finite-state compiler based on weighted compound segments, as described by Schiller [13], gave priority to splits with the least number of segments and compound segments with the highest frequency in a training set. A German decompounder was developed by Alfonseca, Bilac

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 251–258, 2011.

and Pharies [3] comparing different weighting schemes like frequency and probability-based methods and a support vector machine. A frequency-based method was introduced by Koehn and Knight [8], using the geometric mean of compound parts in a corpus, assuming that the more frequently a word occurs in a corpus, the more likely it is to form part of a compound word. The weaknesses of methods like these include incidental high frequency in a corpus (frequency-based) and the preference of minimum number of parts (probability-based) [2].

String-fitting also seems to have been experimented with widely. Monz and De Rijke [10] used a recursive decompounding procedure that greedily chooses the smallest substring of the word that belongs to a lexicon from left to right. This procedure could not handle words with several possible splits. A form of longest common substring (LCS) comparison was applied by Brown [4] to identify corresponding words in German and English texts in the medical domain. Pilon, Puttkammer and Van Huyssteen [11] used, among other techniques, LCS from the beginning and end of words for decompounding Afrikaans words, but results were disappointing.

The algorithm discussed in this paper uses string comparison, but in contrast with conventional LCS where the longest strings from the start and/or the end of words are used, all common strings from the start and end of words are considered in the decompounding process.

## 2 Development of the Algorithm

Since compound words are created by joining existing words, it makes sense to use a list of existing words to decompound Afrikaans words. A lexicon of 182 433 words was therefore compiled from several electronic sources, forming the basis of the decompounding algorithm.

### 2.1 Basic Principles

Initially the complete lexicon is used as reference list (RL). For a word of length $wl$ strings of length $2, 3, \ldots, wl - 2$ from the start (and end) of the word are respectively compared with RL. All corresponding words at the start (and end) of the word are extracted and placed in the "start (and end) word array", called SW (and EW). To determine where a word should be split, words from SW and EW are mutually combined and whenever the combined length ($cl$) is equal to $wl$, the word is split. The following examples illustrate the concept:

- For the word *slaapkamer* (bedroom) with $wl = 10$ the subwords extracted from RL are the following:

| | SW | EW | |
|---|---|---|---|
| (*sleep*) | slaap | kamer | (*room*) |

  Since $cl = 10 = wl$ the word is split as *slaap-kamer*.

- For *rugbytrui* (rugby jersey) with $wl = 9$ the subwords extracted are

|  | SW | EW |  |
|---|---|---|---|
| (*rough*) | ru | ui | (*onion*) |
| (*back*) | rug | trui | (*jersey*) |
| (*rugby*) | rugby |  |  |

  Possible combinations are *ru-ui*, *ru-trui*, *rug-ui*, *rug-trui*, *rugby-ui* and *rugby-trui*. Only *rugby-trui* has $cl = 9 = wl$, which is the correct split.

- For *aasvoëlnes* (vulture's nest) with $wl = 10$ the subwords extracted are

|  | SW | EW |  |
|---|---|---|---|
| (*bait*) | aas | nes | (*nest*) |
| (*vulture*) | aasvoël | voëlnes | (*bird's nest*) |

  Both *aas-voëlnes* and *aasvoël-nes* are valid since $cl = 10 = wl$. Combining these gives the word fully decompounded as *aas-voël-nes*.

## 2.2 Problem Areas

**Simple Words Split.** Simple words were split due to certain short words in RL, for example *aal* (eel), *wens* (wish), *bed* (bed), *ertjie* (pea) causing the errors shown in Table 1.

**Table 1.** Simple words split

| Meaning | Word | Divided | Meaning |
|---|---|---|---|
| *aloe* | aalwyn | aal-wyn | *eel wine* |
| *besides* | benewens | bene-wens | *legs wish* |
| *calming* | bedarend | bed-arend | *bed eagle* |
| *small room* | kamertjie | kam-ertjie | *comb pea* |

This problem was addressed by removing problem words like these from RL and, depending on their effect on other words in the lexicon, restricting them to the start or end of words. Lists were created for words restricted to the start of words (SL) and to the end of words (EL). For example, the word *bed* often caused faulty splitting at the start of words (*bed-agsaam*, *bed-erflik*, *bed-raad*). When it was removed from RL, about 60 splits were missed, but errors were avoided. At the end of words *bed* did not cause any errors, so it was inserted in EL. On the other hand the word *ertjie* caused errors at the end of words (*kam-ertjie*, *hang-ertjie*), while it did not cause errors at the start of words. It was therefore removed from RL and inserted in SL.

**Hyphens on Both Sides of Consonants.** This phenomenon occurs when two different valid splits according to word length are combined. For example, the word *toeroes* (to get covered in rust) had both *toe-roes* (the correct split) and *toer-oes* as valid splits and when they were combined the result was *toe-r-oes*. More examples are *seun-s-kool*, *tee-r-oos*, *verdikking-s-laag*. To solve this problem, the relevant words were, like before, evaluated and either completely removed or inserted in SL or EL.

**Compound Words Undivided.** The reasons for compound words not being divided were identified as described below:

(a) *Linking morphemes* (LM) are often used in Afrikaans to form compound words. This caused the combined length of start and end words not to correspond with the word length. For example, the word *stadslewe* (city life) has $wl = 9$, while the subwords *stad* (city) and *lewe* (life) that were extracted from RL have $cl = 8$. To make provision for linking morphemes the following condition is added to the algorithm: *if* $cl = wl - n$*, for* $1 \leq n \leq 3$ *and the additional letter(s) correspond(s) with a linking morpheme, the word is split after the morpheme*. Table 2 shows the LMs included in the algorithm. Single-letter LMs like *n* and *r* cause splitting errors and were excluded.

**Table 2.** Linking morphemes

| $n$ | Morpheme | Example | Meaning |
|---|---|---|---|
| 1 | s | gebied(s)waters | *territorial waters* |
|  | e | leeu(e)moed | *lion's courage* |
| 2 | ns | ete(ns)tafel | *dining table* |
|  | er | geselsend(er)wys | *in a chatting way* |
| 3 | ens | bejammer(ens)waardig | *pitiable* |

(b) *Inflections* like plurals, diminution and past tense are indicated by affixes in Afrikaans. The lexicon was compiled from electronic corpora and doesn't contain all inflective forms of words, causing compounds containing such inflections not to be split.

The word *spieraanhegtinge* (muscle insertions) was not split since RL does not contain *aanhegtinge*. It does however contain *aanheg* which is inflected by the suffix *tinge*. This is addressed by temporarily removing the suffix and splitting the rest of the word to get *spier-aanheg* and then replaced the suffix to obtain *spier-aanhegtinge*.

The prefix *ge* is used to indicate past tense, for instance the past tense of *bladlees* (sight-read) is *gebladlees*. RL does not contain *geblad* and the word was not split. When the prefix *ge* is temporarily removed, the rest is split as *blad-lees*, and the result is *geblad-lees* when the prefix is replaced.

In Afrikaans the spelling of words often changes with inflection, for instance *onderstreep* (underline) becomes *onderstreping* (underlining) which is not found in RL, causing a word like *fraseonderstreping* (underlining of phrases) not to be split. Although this is a frequent problem, it doesn't cause hyphenation errors and it is left for further research.

(c) *Technostems* such as *geo, elektro, fisio,* are not found in RL. Words containing them were therefore never split. Although one would not find these pseudo-words on their own in texts, they have meaning and in many cases it makes sense to split such words.

The algorithm only splits words where technostems are combined with words in RL, for example *geo-magneties* (geomagnetic), *hidro-geologie* (hydrogeology) and *tegno-politieke* (technopolitical). Since technostems are always found at the beginning of words they were inserted in SL.

(d) *Complex compounds* consisting of three or more words were often left undivided since the longest words in SW and EW do not cover the complete word. For example, for *skaaphondhanteringskompetisie* (sheep dog handling competition) with $wl = 29$, the longest common strings (LCS) from RL are *skaaphond* (SW) and *kompetisie* (EW) with $cl = 19 < 29$.

To address this problem the LCS from the start (*skaaphond*) is temporarily removed and if the rest of the word is split (*hanterings-kompetisie*) the removed part is replaced and a hyphen is inserted (*skaaphond-hanterings-kompetisie*). If the rest cannot be split, the same process is repeated with the LCS from the end. If this also doesn't result in a split, both the LCS from the start and the end are removed and the rest is split. For example, for *waterkultuurvoedingsmengseltablette* (water culture feeding mixture tablets), the LCS from the start is *waterkultuur* and from the end *tablette*. Removing these one after the other does not result in a split, but when both are removed (*voedingsmengsel*) is split resulting in *waterkultuur-voedings-mengsel-tablette*.

Both examples discussed above are not fully decompounded, since *skaaphond* (sheep dog) and *waterkultuur* (water culture) are also compounds. When a word has been split successfully, each subword is split until no further splits occur. The final results are *skaap-hond-hanterings-kompetisie* and *water-kultuur-voedings-mengsel-tablette*.

(e) *Short words removed from RL* were inserted in SL or EL according to their influence on other words (see Sect. 2.2). As a final step in the algorithm words in SL and EL are considered. For instance, *sterfbedwoorde* (deathbed words) was split as *sterfbed-woorde*. When SL and EL were taken into account, *sterf* was found in RL and *bed* in EL and the result was *sterf-bed-woorde*.

## 3 The Decompounding Algorithm

The algorithm was programmed in PERL. It consists of a main program that calls four subroutines for the decompounding process. The data used for decompounding consist of the reference list RL (181 362 words), lists of words restricted to the start and end of words SL (510 words) and EL (376 words), and lists of prefixes PFL (29) and suffixes SFL (171).

A word presented to the algorithm is divided at hyphen(s) (if present) and the word, or each subword, is sent through the following decompounding loop: (i) Start with the basic split using only RL and linking morphemes; (ii) if the word is not split, also consider affixes; (iii) if the word is not split, remove LCS from start and/or end and split the rest; (iv) if the word is not split, consider short words in SL and EL.

If the word has passed through all subroutines without being split, it is regarded as a simple, undivisible word and the output is the original word. If it is split in one of the subroutines, each subword is sent through the loop until the word is not split any further. The output is the fully decompounded word.

## 4 Performance Measures and Results

The compounds in the lexicon of 182 433 were split manually. This list (CS) is regarded as mostly correct, since errors were identified and corrected during the development process. The algorithm was applied to the lexicon and the results were compared with CS. Statistics regarding complete words as well as splitting opportunities[1] were recorded. The outcomes that were considered with regard to complete words (and splitting opportunities) are the following:

- Compounds correctly split (hyphen correctly inserted) – Correct positive ($C_p$);
- Simple words left unsplit (hyphen correctly omitted) – Correct negative ($C_n$);
- Words with wrong splits (hyphen incorrectly inserted) – Fault positive ($F_p$);
- Compounds with missed splits (hyphen missed) – Fault negative ($F_n$).

The outcomes are shown in Table 3.

**Table 3.** Outcomes of the algorithm

| | | Complete words | | Splitting opportunities | |
|---|---|---|---|---|---|
| | | Split | No split | Hyphen | No hyphen |
| Algorithm | Split | $C_p$<br>89 620 (49,12%) | $F_p$<br>87 (0,05%) | $C_p$<br>100 049 (4,92%) | $F_p$<br>87 (0,00%) |
| | No split | $F_n$<br>17 619 (9,66%) | $C_n$<br>75 107 (41,17%) | $F_n$<br>18 785 (0,92%) | $C_n$<br>1 913 491 (94,15%) |

The measures[2] used to evaluate the algorithm's performance are described below and the results are shown in Table 4.

(a) *Accuracy* – the probability of correctly splitting compounds and leaving simple words unsplit (or correctly classifying splitting opportunities as *insert hyphen* or *do not insert hyphen*), with

$$A = \frac{C_p + C_n}{C_p + C_n + F_p + F_n}.$$

(b) *Precision* – the probability of correctly splitting compounds (or correctly inserting a hyphen), with

$$P = \frac{C_p}{C_p + F_p}.$$

(c) *Recall* – the probability of correctly splitting compounds and not leaving them unsplit, with

$$R = \frac{C_p}{C_p + F_n}.$$

[1] The lexicon contains 2 032 412 positions between letters which are regarded as possible splitting opportunities.

[2] Adapted from performance measures used for information retrieval [9].

(d) $F$ count – the traditional, or balanced, weighted harmonic average of precision and recall, with

$$F = 2 \times \frac{P \times R}{P + R}.$$

**Table 4.** Performance of the algorithm

| Measure | Complete words | Splitting opportunities |
|---|---|---|
| Accuracy | 90,3% | 99,1% |
| Precision | 99,9% | 99,9% |
| Recall | 83,6% | 84,2% |
| $F$ count | 91,0% | 91,4% |

The algorithm clearly performs well with 99,9% precision on complete words, meaning the probability of compounds being split incorrectly is 0,1%. Accuracy on splitting opportunities is 99,1% which means that the probability of inserting or omitting a hyphen wrongly is less than 1%. Relatively low recall ($\approx$ 84%) is due to words being left undivided, mainly due to (i) short problem words that were removed from RL, and (ii) inflections with changed spelling that are not present in RL.

## 5 Conclusions

The purpose of this paper is to report on a recursive decompounding algorithm that has been developed for Afrikaans. Evaluation of the algorithm shows that employing complete string comparison from the beginning and end of words, and recombining subwords to comply with word length constraints gives good results on the lexicon available.

Success is illustrated by long compounds split completely (e.g. *obligasie-mark-termyn-handels-kontrakte* and *geel-bek-neus-horing-voël*) as well as long simple words left undivided (e.g. *kompartementalisering* and *verdiskonteerbaar*).

Suggestions for further research include the application of traditional machine learning techniques like neural networks and decision trees to this problem, and the combination of the algorithm described here with such techniques to possibly improve the recall.

## References

1. Adda-Decker, M., Adda, G., Lamel, L.: Investigating text normalization and pronunciation variants for German broadcast transcription. In: ICSLP, pp. 266–269 (2000)
2. Alfonseca, E., Bilac, S., Pharies, S.: Decompounding query keywords from compounding languages. In: ACL 2008: HLT, pp. 253–256 (2008)
3. Alfonseca, E., Bilac, S., Pharies, S.: German Decompounding in a Difficult Corpus. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 128–139. Springer, Heidelberg (2008)

4. Brown, R.D.: Corpus-driven splitting of compound words. In: TMI-2002, pp. 616–624. ACL (2002)
5. Fick, M., Swanepoel, C.J.: Afrikaanse Lettergreepverdelingspatrone. Suid-Afrikaanse Tydskrif vir Natuurwetenskap en Tegnologie (2010)
6. Fritzinger, F., Fraser, A.: How to Avoid Burning Ducks: Combining Linguistic Analysis and Corpus Statistics for German Compound Processing. In: MATR, pp. 224–234. ACL (2010)
7. Koehn, P., Arun, A., Hoang, H.: Towards better Machine Translation Quality for German–English Language Pairs. In: Third Workshop on Statistical Machine Translation, pp. 139–142. ACL (2008)
8. Koehn, P., Knight, K.: Empirical methods for compound splitting. In: EACL, 187–193. ACL (2003)
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
10. Monz, C., De Rijke, M.: Shallow Morphological Analysis in Monolingual Information Retrieval for Dutch, German, and Italian. In: Peters, C.A., Braschler, M., Gonzalo, J., Kluck, M. (eds.) CLEF 2001. LNCS, vol. 2406, pp. 262–277. Springer, Heidelberg (2002)
11. Pilon, S., Puttkammer, M.J., Van Huyssteen, G.B.: The development of a hyphenator and compound analyser for Afrikaans. Literator (2008)
12. Popović, M., Stein, D., Ney, H.: Statistical machine translation of German compound words. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 616–624. Springer, Heidelberg (2006)
13. Schiller, A.: German compound analysis with wfsc. In: Finite State Methods and NLP (2005)

# Reliable Detection of Important Word Boundaries Using Prosodic Features

Caroline Kaufhold, Georg Stemmer, and Elmar Nöth

Universität Erlangen-Nürnberg, Lehrstuhl für Mustererkennung (Informatik 5)
Martensstraße 3, 91058 Erlangen, Germany
noeth@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de

**Abstract.** Natural input dialog systems for entering address data in modern GPS units demand a significantly more robust extraction of slot information. In previous work, prosody was used to detect phrase boundaries (PB) which separate particular address parts. Only input without filler words was used. In this work, carrier sentences are allowed. A boosting approach provides twenty strong prosodic features which model the characteristic of PBs. We introduce the concept of a prosodically marked word boundary (PMB), which enables a better location of the provided information in natural input. Our results on a dataset of 5883 input samples reveal that about 67 % of the found PBs indicate a PMB, while most of the remaining boundaries occur within compound words.

**Index Terms:** prosody, phrase boundary detection, multi-slot input modality.

## 1 Introduction

Nowadays, GPS navigation systems which provide speech input become more and more common. In order to provide service, particular information like the destination address has to be known. This is done by filling several slots which represent each a particular part of the address like *city name* or *street name*. In general, one slot is filled after another such that the user is forced to a specified order in which the information can be entered. An example dialog of such a restricted system would be:

sys: *Please say the city name*
user: *Erlangen-Bruck*
sys: *Please say the street name*
user: *Gerhart-Hauptmann-Straße*
sys: *Please say the street number*
user: *17*

A major goal of our work is to provide a speech input modality which also accepts more natural spoken input in order to design a more user-friendly and intuitive user interface. In our previous work [1] we looked at a first step towards unrestricted input: Multi-slot filling. The dialog above could then be as follows:

sys: *Please say the address*
user: *Erlangen-Bruck, Gerhart-Hauptmann-Straße 17*

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 259–267, 2011.

Of course, the combinatorial explosion of street and city names will pose a problem to the speech recognition system which has to run on a small-footprint processor. This is why we tried to predict the boundary between the two slots *city name* and *street name* with prosodic information independent of the speech recognizer. In previous work, this boundary was referred to as Phrase Boundaries (PB), since slot information may consist of one word or even more and was therefore treated as phrases. Due to more complex input utterances which contain also filler words or whole carrier sentences, these boundaries are, however, called Slot Boundaries (SB) within this work. Just using prosodic information we achieved an F-measure of .83, taking the relative position of the predicted boundary within the utterance into account we achieved .93 F-measure. The purpose of this paper is to see what happens when we take spoken input in GPS navigation system dialogs one step further towards a natural dialog model and allow an even less restricted input, i.e., the user is confronted with a *"How may I help you"* system [2]. We wanted to see, where the prosodically most prominent boundaries that are automatically classified with our system are located in the unrestricted utterances.

The rest of the paper is organized as follows: In Chapter 2 we introduce the speech database, in Chapter 3 we look at the word boundaries of our utterances and explain how they are automatically detected. In Chapter 4 we perform a detailed analysis of the classification results. The paper ends with a summary and outlook.

## 2 Speech Database

Two different speech databases, *Er-Car-List* and *Er-Car-Conv*, were used in this work for the computation and evaluation of word boundaries. Each represents a specific input modality which provides different ways of producing addresses in German in order to enter them into a navigation system.

The recordings belonging to *Er-Car-List* are simple examples for multi-slot filling utterances. Two major slots of information are filled in each recording: *city name* and *street name*. The latter could optionally contain a *street number*, i.e., the *street number* was not considered a slot on its own but an optional part of the slot *street name*. *Er-Car-List* contains recordings from 109 speakers. A subset of 97 speakers (48 female and 49 male) produced a set of 150 different addresses of the type *city name, street name* and, in addition, 50 different addresses which also contained the *street number* (*city name, street name, street number*). The remaining 12 speakers (3 female and 9 male) produced the 150 input signals the other way round, *street name, city name*, and additionally 50 addresses containing also the *street number* (*street name, street number, city name*). Thus the following utterances occurred: *City SB Street*; *City SB Street, Number*; *Street SB City*; and *Street, Number SB City*. The utterances were checked for completeness and correctness by a human labeler. In addition, phoneme hypotheses as well as an estimation of the SB position (based on forced time alignment) were computed for all recordings. *Er-Car-List* is the database that was used in [1] for train and test.

The second speech database, *Er-Car-Conv*, provides recordings resembling a natural input mode. Recordings consist of content words (slot information for the slots *city name*, *street name* and *street number*) and non-content words which are part of the carrier sentence. *Er-Car-Conv* can be subdivided into three major classes: *conversational*, *command-like*, and *strictly formatted* input utterances. Recordings of the latter class exclusively contain random subsets of content words which are valid addresses without any filler words. Examples range from a complete address (multi-slot) to single address components (one slot). Recordings belonging to the command-like input class additionally contain keywords like *destination* or *stopover*. Finally, the most natural input class which contains conversational recordings permits carrier sentences like *Please, navigate me to . . . in . . .* . Thus, the number of content words – *city name*, *street name*, *street number* – which is encoded in the particular carrier sentences varies. Prior to recording, the speakers were instructed to read out given directions from a screen as if they were using their GPS unit in real life. However, the utterances were not checked for completeness and correctness by a human labeler. For each of the 48 speakers (15 female and 33 male), *Er-Car-Conv* contains 130 to 200 recordings belonging to the three input modes *conversational*, *command-like*, or *strictly formatted*. For each recording, a word segmentation based on the forced time alignment with the reference text was computed. A phonetic transcription is currently in progress.

## 3 Prosodically Marked Boundaries

In our previous work, the slot boundaries (SB) were exclusively boundaries between the slots in lists for multi-slot filling. We assumed that these word boundaries would be prosodically marked stronger than the word boundaries within the slot content (boundaries which separate compound words, pairs of words as well as chunks of semantic information which may contain several words). Twenty strong prosodic features were computed using an automatic feature selection procedure [1,3] in order to characterize the SB regarding all possible combinations of content words. However, more natural input generally consists of carrier sentences which encode a variable number of slot information. Consequently, there are also boundaries between two non-content words or between a content word and a non-content word and vice versa. For that reason the boundary between a content word and its direct predecessor or successor is of particular interest. Since this approach considers boundaries which are prosodically marked they are referred to as “Prosodically Marked Boundaries” (PMB). For instance, within the input utterance *Please, navigate me to to Gerhart-Hauptmann-Straße 17 in Erlangen-Bruck* the PMB in question are *Please, navigate me to* PMB *Gerhart-Hauptmann-Straße 17* PMB *in* PMB *Erlangen-Bruck*. However, especially with difficult names and a more emphatic prosody, the user might consider name parts like *Straße* (street) as a carrier word and prosodically mark a boundary *Gerhart-Hauptmann* PMB *Straße 17* or separate the city (Erlangen) from the district (Bruck), i.e., *Erlangen* PMB *Bruck*. Therefor we want to investigate how well the set of strong prosodic features which has been computed for detecting SB performs on finding PMB within more general input data.

The segmentation was done independent of existing speech recognition systems and prosodic information was computed using the Erlangen Prosody Module[4][5] as described below.

### 3.1 Erlangen Prosody Module

The prosodic features are computed using the Erlangen Prosody Module (EPM). The EPM was implemented at the Chair of Pattern Recognition at the University of

Erlangen-Nuremberg in the course of the Verbmobil project [6] [5]. Values of fundamental frequency ($F_0$) and short time signal energy are computed for short speech frames with a frame rate of 10 ms. These are tied to the voiced/unvoiced decision [7]. In order to examine the suprasegmental characteristics of prosodic phenomena a segmentation is then performed by using the voiced/unvoiced decision. For each voiced segment, features like the maximum $F_0$ value are computed for the segment itself and the segment and its neighboring segments. All features are computed within a context of at most $\pm 2$ voiced segments. Finally, a vector of 187 prosodic features is computed for each voiced segment. A detailed description of the feature computation is given in [8].

### 3.2 Slot Boundary Model

Multi-slot filling input utterances contain information of several slots. The respective parts of the signal have to be recognized as content words and have to be assigned to the corresponding slots. In the recordings of *Er-Car-List*, slot information is simply concatenated and entered as a list. In addition, items in spontaneously produced lists are generally realized such that they are perceptually salient (as described in [9,10]). Boundaries separating content words like *street* or *city name* are probable to occur at speech pauses. During the voiced/unvoiced decision they are recognized as voiceless speech parts. The prosodic changes describing the transition from one information chunk to the next can be modeled using prosodic information from the voiced speech parts which are surrounding an unvoiced period. The approach followed in our previous work was therefore to concatenate the prosodic feature vectors of two consecutive voiced segments, yielding a vector with 374 elements. Since this vector is rather large and in particular not advantageous in terms of computation, 20 strong prosodic features are selected based on a turn-based selection algorithm in order to reduce dimensionality. The particular prosodic features are selected according to their contribution to finding SB which is explained in more detail in [1]. Table 1 shows some of the most important features selected by the algorithm which describe best the boundary between the major information slots: *city name* and *street name* which is optionally extended by a *street number*. In addition to durational features which indicate the application of final lengthening, features describing the progression of fundamental frequency ($F_0$) are used. Even shimmer as a micro-prosodic feature is selected. However, the most significant information is given by the length of silence in the current unvoiced segment. All remaining features are energy features.

**Table 1.** The most important selected features for detection of SB in recordings of *Er-Car-List*

| prosodic feature expresses |
|---|
| length of silence in current segment |
| mean $F_0$ within left (voiced) segment |
| duration of left and second left segment |
| duration of left segment |
| relative position of max $F_0$ within right segment |
| variance of $F_0$ period shimmer of second left segment |

### 3.3 Detecting Prosodically Marked Boundaries

The question we deal with in this work is: Do the results found for detecting SB transfer to natural input?

Twenty strong prosodic features were chosen according to their contribution on detecting the boundary between major information slots within recordings of *Er-Car-List*. More natural input contains non-content words and does, thus, not contain such clear boundaries. In order to detect a content word encoded within non-content words, specific boundaries, PMB, indicating its beginning and ending have to be detected. The SB Model was applied on the natural input utterances of *Er-Car-Conv* such that for each feature vector only the 20 selected strong prosodic features were considered. The recordings were then classified by a classifier which was trained on the data of *Er-Car-List*. The classification results are discussed in the following Section.

For classification, a *J48* Decision Tree is trained in a supervised manner on the recordings of *Er-Car-List*. For testing, the data of *Er-Car-Conv* is used. In order to detect PMB on the basis of prosodic features describing SB, only the strong prosodic features are considered.

## 4 Detailed Analysis

The feature vectors computed from recordings of *Er-Car-Conv* each contain values for the 20 selected strong prosodic features. Given a classifier trained on the recordings of *Er-Car-List* each vector gets a probability assigned according to its extent of resembling a potential PMB. In the following, these classified boundaries are analyzed with special regard to the type of the neighboring words in question - content or non-content words.

*Er-Car-Conv* contains 5893 recordings whereof 878 consist of only one word. The maximum number of words is 12 which applies for 5 recordings. There are 15201 potential PMBs between word pairs of all recordings. Each unvoiced speech part within a

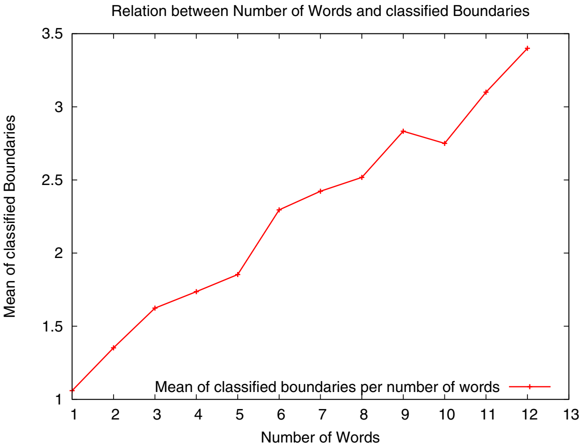


**Fig. 1.** Number of classified boundaries belonging to recordings of a given number of words

recording is represented by a feature vector. For the complete dataset, an overall number of 29952 feature vectors is computed. Classification results in 8477 of 29952 feature vectors which are classified as PMB and 21475 as no PMB. All in all there are 1232 recordings which do not contain a classified boundary at all. A detailed analysis of the number of classified boundaries with respect to the type of the input utterance - strictly formatted, command-like or conversational - is given below.

Clearly, the number of unvoiced segments that are classified as a boundary, should depend on the length of the utterance. The almost linear relationship between the number of words in the reference text vs. the number of unvoiced segments that were classified as boundary is shown in Figure 1. The longer the utterance is in terms of words, the more potential boundaries are classified as PMB.

### 4.1 Recordings Containing No Classified Boundaries

There are 1232 recordings of all 5893 recordings of *Er-Car-Conv* within which no unvoiced part of speech was classified as a boundary. In the case of recordings which consist only of a single word it is assumed that there are no classified boundaries. 51% of the recordings which do not contain a single PMB are recordings consisting of only one word. At the same time, they account 71% of all recordings which only contain a single word. Almost all of the remaining 29% recordings which contain single word, are compound words within which only one boundary is detected. The relationship between the number of classified boundaries and the words within the concerned recordings is shown in more detail in Figure 2. The remaining recordings within which no boundary was classified at all are mainly compound words like “Schwarzer Weg” or a combination of street name and street number. Since these utterances do not contain a boundary in the sense of SB they are not classified as such.

### 4.2 Recordings Containing at Least One Classified Boundaries

Within 38% recordings of the *Er-Car-Conv* database a single unvoiced speech part was classified as PMB. In Figure 2 the relationship between these recordings and their number of words (dark dotted line) is depicted. The recordings consisting of two to three words make up 56% of all recordings containing one classified boundary. The majority of the utterances containing only two words are compound words. The utterances which contain three words are, however, mostly a combination of street name and street number. It is interesting to see that in the latter case the boundary between street name and street number is recognized as a SB. Since these recordings only consist of those two content words, street number is likely to be treated as a further content word and is emphasized as such.

The light dotted line in Figure 2 depicts the distribution of recordings containing more than one classified boundary for different lengths of recordings in terms of number of words. They are 41% of all recordings of the *Er-Car-Conv* database which were taken for testing. Utterances comprising a street number in combination with content words mainly contain more than one classified boundary. That is the case for almost all utterances containing three to four words. Recordings containing only two words mainly consist, however, of content words.

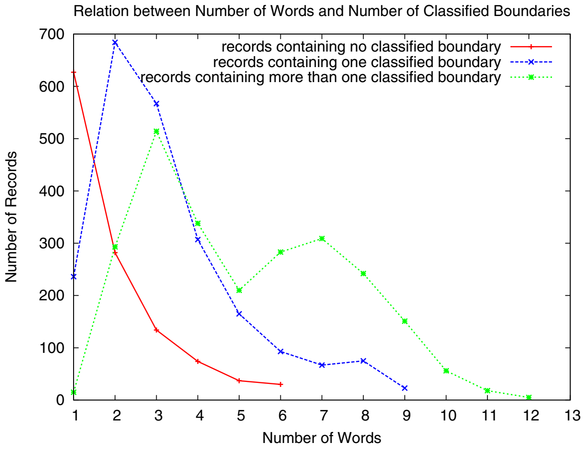


**Fig. 2.** Number of words in the recordings which contain at least one classified boundary or no classified boundary vs. number of recordings

### 4.3 PMBs between Two Content Words

In the following the different types of boundaries which were recognized by the classifier trained in order to detect SBs, are discussed. The words surrounding the unvoiced speech part which was classified as PMB are therefore examined. There are three different kinds of boundaries: between two content words (street and city name; SB), within a single content word, between non-content words and content words. There are 8477 classified PMBs within all recordings of *Er-Car-Conv*. Boundaries between non-content words and content words as well as between only non-content words are the majority of PMBs with 67%. The remaining 33% are almost all between two content words.

The group of PMBs between two content words can be subdivided into two subcategories: 60% of them resemble boundaries in the sense of SB and 40% occur within a content word which comprises more than one word. The latter subcategory is of particular interest because more than half of the word combinations are compound words like “Dreiweiler Straße” which contain the word “Straße” (German for “street”). Since street names containing the word “street” particularly differ in the part which precedes the word “street” it is likely that this part is particularly emphasized. Furthermore, it can be assumed that “street” is considered as a carrier word in order to particularly stress this boundary such that it is realized and classified as SB, respectively.

### 4.4 PMBs between a Content Word and a Non-content Word

PMBs which describe the transition from a non-content word to a content word like in “in PMB Erlangen” or vice versa make up 57% of all classified boundaries. One fourth describes the boundary between a content word followed by a street number. By contrast, only 8% of PMBs describe a boundary between a street number and a content word. In German addresses the street name is generally followed by the street number. Since the combination of street name and number the other way around is rather unusual, the realization of this kind of boundary is probably not distinctive enough. One third of all PMBs between content and non-content words describe the transition from a non-content word like "in" or a command-word like "destination" to a content word. Take for instance the utterance "ziel PMB NOSCHKOWITZ". The PMB following the command-word "ziel" (German for "destination") indicates that there is a more prominent word, a content word, following. The remaining PMBs describe boundaries between a content word followed by a non-content word. The majority of non-content words are prepositions like "in" or "to" or words containing the word "number".

The class of PMBs which describe boundaries between exclusively non-content words only make up 17% of all 8477 classified boundaries. In almost all cases, one of the two words which precede or succeed the boundary is either a command-word like "starten" or "please". The following phrase, e.g. "Please, can you tell me the way to ..." is quite commonly used in everyday conversations. It can be assumed that it is also used in human-machine interaction.

## 5 Summary and Outlook

In this work, we applied a classifier trained on SB in a restricted database (multi-slot filling in a car navigation scenario) to more conversational input in the same scenario. We presented a detailed analysis of how the automatically predicted boundaries of an utterance coincide with different classes of word boundaries. The results indicate that 57 % of the predicted boundaries are word boundaries, 33 % of the predicted boundaries are between content words, 60 % of which denote SBs, and 40 % of these boundaries occur within a content word. Furthermore, the relation between the number of words within a recording and the number of predicted boundaries is almost linear.

Speech recognition systems of applications like GPS navigation units have to become more robust against natural input containing carrier sentences and further filler words in order to enhance usability. Our results indicate that a prosodically based boundary predictor can be an important knowledge source and that the prosodic marking of slot boundaries in sequences of semantic slots generalizes to PMB in more conversational utterances (albeit not perfectly). Our current work concerns the enhancement of the database and the hand labeling of the recordings in order to make a more detailed error analysis and significantly improve our classification results.

## References

1. Kaufhold, C., Nöth, E.: Using prosodic features for predicting phrase boundaries. In: Speech Prosody (2010)
2. Gorin, A.L., Riccardi, G., Wright, J.H.: How may I help you? Speech Communication 23, 113–127 (1997)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11 (2009)

4. Kießling, A.: Extraktion und Klassifikation prosodischer Merkmale in der automatischen Sprachverarbeitung. PhD thesis, Pattern Recognition Lab, University of Erlangen-Nuremberg (1996)
5. Batliner, A., Buckow, J., Niemann, H., Nöth, E., Warnke, V.: The prosody module. In: Wahlster, W. (ed.) Verbmobil: Foundations of Speech-to-Speech Translations, pp. 106–121. Springer, Heidelberg (2000)
6. Wahlster, W.: Verbmobil: Foundations of Speech-to-Speech Translation. Springer, Germany (2000)
7. Maier, A., Hönig, F., Zeissler, V., Batliner, A., Körner, E., Yamanaka, N., Ackerman, P., Nöth, E.: A language-independent feature set for the automatic evaluation of prosody. In: Interspeech (2009)
8. Hönig, F., Batliner, A., Weilhammer, K., Nöth, E.: Islands of failure: Employing word accent information for pronunciation quality assessment of English l2 learners. In: Interspeech (2009)
9. Selting, M.: Lists as embedded structures and the prosody of list construction as an interactional resource. Journal of Pragmatics 39, 483–526 (2007)
10. Shukla, M., Nespor, M., Mehler, J.: An interaction between prosody and statistics in the segmentation of fluent speech. Cognitive Psychology 54, 1–32 (2007)

# Rule-Based Triphone Mapping for Acoustic Modeling in Automatic Speech Recognition

Sakhia Darjaa[1], Miloš Cerňak[1], Štefan Beňuš[1,2],
Milan Rusko[1], Róbert Sabo[1], and Marián Trnka[1]

[1] Institute of informatics, Slovak Academy of Sciences
Dúbravská c. 9, 845 07 Bratislava, Slovakia
[2] Department of Eng. and Am. Studies Constantine the Philosopher University, Nitra, Slovakia
{sachia.darzagin,milan.rusko,
robert.sabo,milos.cernak,trnka}@savba.sk, sbenus@ukf.sk

**Abstract.** This paper presents rule-based triphone mapping for acoustic models training in automatic speech recognition. We test if the incorporation of expanded knowledge at the level of parameter tying in acoustic modeling improves the performance of automatic speech recognition in Slovak. We propose a novel technique of knowledge-based triphone tying, which allows the synthesis of unseen triphones. The proposed technique is compared with decision tree-based state tying, and it is shown that for bigger acoustic models, at a size of 3000 states and more, a triphone mapped HMM system achieves better performance than a tree-based state tying system on a large vocabulary continuous speech transription task. Experiments, performed using 350 hours of a Slovak audio database of mixed read and spontaneous speech, are presented. Relative decrease of word error rate was 4.23% for models with 7500 states, and 4.13% at 11500 states.

**Keywords:** automatic speech recognition, acoustic modeling, model tying.

## 1 Introduction

Statistical modeling dominates in current speech technology. In automatic speech recognition (ASR), rare triphones are tied on the model [1] or the state [2] level, and such context modeling based on either data-driven or decision tree clustering significantly improves the recognition performance. It was already shown that the state tying system consistently out-performs the model clustered system.

Phonetic decision tree-based state tying utilizes the knowledge of phonetic classes determining contextually equivalent sets of HMM sets. Facing the challenge of recovering linguistic information in acoustic modeling, which is one of the area for future ASR research specified by [3], we re-visited the process of building the HMM system for Slovak language, showing that our proposed phonetic rule-based triphone tying HMM system outperforms the tree-based state tying HMM system. The performance gain is achieved with effective triphone mapping, and its latent use in the process of building an HMM system.

The remainder of the paper is structured as follows. In the next Section 2 we introduce rule-based triphone mapping, which we apply to a large-vocabulary continuous

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 268–275, 2011.

speech transcription task in the experimental part of the paper in Section 3. Finally in Section 4 we discuss achieved results.

## 2 Rule-Based Triphone Mapping

Triphones are context phonemes (basis phoneme $P$ with the left and right context: $P_{\mathrm{left}}$–$P$+$P_{\mathrm{right}}$). Most triphones are rare and it is not possible to train them robustly. We therefore map rare triphones to more frequent triphones that are much better trained. Thus we constrain contextual information, based on context similarity.

The process of building a triphone mapped HMM system has 4 steps:

1. Training of monophones models with single Gaussian mixtures
2. The number of mixture components in each state is incremented and the models are trained
3. The state output distributions of the monophones are cloned, triphone mapping is applied
4. The triphone tied system is trained again

Unlike the process of building a tied state HMM system [2], monophone models are trained with multiple Gaussian mixtures, and subsequently, state output distributions are cloned for triphone models initialization with a latent application of the triphone map. In a tied state HMM system, cloning and state clustering is done on single Gaussian mixtures, and then the number of mixture components is incremented.

First, the selection of most frequent triphones is performed. Triphones are sorted according to occurrence and a limit is determined. The typical limit from 400 to 800 occurrences is used for databases extending hundred hours. Top $N$ (usually from 2000 to 3500), most frequent triphones, are thus selected from all available contexts. As mapping is not applied to context-free phonemes, such as *sp* and *sil*, they are added to the selection list as monophones. If there are less frequent phonemes that are not represented in the middle part of triphones, these are added to the selection list as well.

### 2.1 Rules for Phonetic Similarity in Slovak

We used a discrete rule-based approach for determining an ordered list of candidate phonemes for each of the 45 target phonemes of Slovak. These candidate lists are ordered based on the phonetic distance from the target phoneme to the candidate phoneme. The ordering process was based on several basic principles that are motivated by general and Slovak-specific phonetic considerations and on strategies for resolutions if the principles are in conflict or if they are not sufficient for uniquely populating the ordered lists. Fig. 1 enlists the ordered 10 (for illustration) candidate phonemes for a set of selected target phonemes.

The principles are violable (in the sense of constraints of Optimality Theory [4], and range from general, such as the preservation of the identity of segment types (consonant, vowel) or the possibility of substitution between vowels and sonorants, to more specific ones, such as the preference for the identity of the manner of articulation and voicing in consonants over the identity in the place of articulation, the preference of preserving vowel height over its frontness, or the preference for the substitution of target

| | | Candidates | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Target** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Syllabic** | a | a | o | r | ɛ | l | u | ɪ̆a | i | a: | ɪ̆e |
| | a: | a: | a | o | ɛ | o: | ɛ: | u: | r | ɪ̆a | ɪ̆e |
| | ɛ | ɛ | i | i: | ɪ̆e | a | o | j | ɪ̆a | ɪ̆u | ʊ̆o |
| | i | i | ɛ | j | n | m | r | u | i: | u: | ɪ̆e |
| | o | o | a | u | ɛ | ʊ̆o | r | ɪ̆e | ɪ̆u | ɪ̆a | l |
| | u | u | o | i | o: | ɪ̆u | r | ɛ | l | i: | a |
| | ɪ̆e | ɪ̆e | ɛ | ɪ̆a | ɪ̆u | ʊ̆o | a | o | u | l | i |
| | l | l | l | r | r: | o | a | ɛ | i | u | ɪ̆a |
| | r | r | r | l | r: | a | o | ɛ | i | u | a: |
| **Consonantal** | b | b | g | d | p | r | j | m | n | l | ɟ |
| | d | d | ɟ | b | g | m | n | dz | z | ɦ | r |
| | dʒ | dʒ | dz | d | ɟ | ɦ | c | s | k | t | ts |
| | f | f | t | p | k | x | dz | z | s | ɦ | ts |
| | g | g | b | d | dz | ɟ | n | m | r | j | ɦ |
| | k | k | t | p | x | ɦ | g | f | d | c | ɟ |
| | n | n | ɲ | m | ɟ | b | d | dz | g | j | r |
| | p | p | t | k | c | f | b | x | r | v | ɟ |
| | s | s | ʃ | f | ts | tʃ | x | t | k | p | c |
| | ʃ | ʃ | tʃ | s | f | ts | dʒ | dz | x | ʒ | p |
| | t | t | p | k | c | f | ts | dz | ɟ | s | tʃ |
| | tʃ | tʃ | ts | f | s | ʃ | z | ʒ | dʒ | k | p |

**Fig. 1.** Ordered canditate list for selected target phonemes

diphthongs with those candidate monophthongs that are identical to the second element of diphthongs. This last strategy is motivated by the fact that Slovak has so called rising diphthongs in which the $2^{nd}$ element is more prominent than the first one.

The heuristic strategies, which filled the gaps or resolved the conflicts after the application of the principles, were based on minimizing the effect of a substitution on the surrounding phonemes taking into account both acoustic and articulatory considerations. For example, /r/ is acoustically the most similar to a schwa-like vowel, thus affects format transitions minimally, and articulatorily involves only a brief tongue tip gesture, thus minimally affecting the tongue body as the main vocalic articulator. Both of these features play an important role in a relatively close proximity (i.e. low rank) of /r/ in the candidate lists for the back vowels /a/ and /i/.

The resulting discrete matrix of partial phoneme confusions thus provides an input for the algorithm that uses the phoneme distance (a position of candidate phoneme in target phoneme row) and subsequently maps each triphone into the closest triphone from the list of selected triphones with the same basis phoneme. The task of triphone mapping consists of separate left context and right context mapping, based on the basic

```
for each triphone Pi-P+Pj (incl. unseen) do:
  for each triphone Pm-P+Pn from the selected triphones with
  the same basis phoneme P do:
    perform left context mapping:
      target_phoneme = Pi
      candidate_phoneme = Pm
      left_context = position of candidate_phoneme in the list
                     belonging to target_phoneme
    perform right context mapping:
      target_phoneme = Pj
      candidate_phoneme = Pn
      right_context = position of candidate_phoneme in the list
                      belonging to target_phoneme
    context_tying_cost = left_context + right_context
  perform triphone mapping:
    Pi-P+Pj is mapped to Pm-P+Pn with minimal context_tying_cost
    if there are more Pm-P+Pn with minimal context_tying_cost do:
      for each Pm-P+Pn do:
        if left_context < right_context do:
          Pi-P+Pj is mapped to Pm-P+Pn
```

**Fig. 2.** Algorithm of triphone mapping. Having a single mapped triphone from the list of all triphones, $P_i$–$P$+$P_j$, and multiple mapping candidate triphones $P_m$–$P$+$P_n$ with the same basis phoneme $P$, the candidate triphone with minimal context tying cost and better left context mapping is selected.

premise that the left context is more important then the right context. Fig. 2 presents the algorithm of context tying for triphone mapping in meta programing language.

## 3 Experiments

The aim of the experiment was to compare tree-based state tying with triphone mapping systems (data-driven state clustering [5] was not considered, as it does not allow synthesis of unseen triphones). Both systems were trained using the same number of Baum-Welch re-estimations, the same number of Gaussian mixtures, and used the same initial set of untied triphones. The tree-based state tying system was trained according to [2] and [6] training procedures. The triphone mapped system was then created according to Sec. 2.

Julius decoder [7] was used as a reference speech recognition engine, and the HTK toolkit was used for word-internal acoustic models training. A set of phonetic questions used in decision trees was taken from the multi-lingual system [8], where the Slovak system achieved state-of-the-art performance when compared to other participating languages. To gain some impression of used questions, Tab. 1 shows the criteria for phonetic grouping used in decision trees.

**Table 1.** The criteria for phonetic grouping used for questions in tree-based state tying in Slovak speech recognition system. Both right (R) and left (L) contexts were considered.

| Vowels | Consonants |
|---|---|
| R,L-short | R,L-sonants |
| R,L-long | R,L-plosives; voiced/unvoiced |
| R,L-monophtongs | R,L-fricatives; voiced/unvoiced |
| R,L-diphtongs | R,L-affricatives; voiced/unvoiced |
| R,L-front | R,L-labial |
| R,L-back | R,L-glottal |
| R,L-open, closed | R,L-lingual |
| R,L-halfopen | R,L-unvoiced |

### 3.1 Data

Experiments have been performed using both read and spontaneous speech databases of the Slovak language. The first database contained 250 hours of gender balanced read speech, recorded from 250 speakers with a Sennheiser ME3 Headset Microphone with an In-Line Preamplifier Sennheiser MZA 900 P. The second database contained 100 hours of 90% male spontaneous speech, recorded from 120 speakers at council hall with goose neck microphones. Databases were annotated using the Transcriber annotation tool [9], twice checked and corrected. Whenever possible, recordings were split into segments not bigger than 10 sec. Our testing corpus contained 20 hours of recordings obtained by randomly selecting segments from each speaker contained in the first read speech database. These segments were not used in training.

A text corpus was created using a system that retrieves text data from various Internet pages and electronic sources that are written in the Slovak language. Text data were normalized by additional modifications such as word tokenization, sentence segmentation, deletion of punctuation, abbreviation expanding, numerals transcription, etc. The system for text gathering also included constraints such as filtering of grammatically incorrect words by spellchecking, duplicity verification of text documents and others constraints. The text corpora contained a total of about 92 million sentences with 1.25 billion Slovak words. Trigram language models (LMs) were created with a vocabulary size of 350k unique words (400k pronunciation variants) which passed the spellcheck lexicon and subsequently were also checked manually. As a smoothing technique the modified Kneser-Ney algorithm was used [10].

### 3.2 Results

First, we trained acoustic models (AMs) using tree-based state tying. By setting 1) the outlier threshold that determines the minimum occupancy of any cluster (RO command), and 2) the threshold of the minimal increase in log likelihood achievable by any question at any node of the decision tree (the first argument of TB command), we trained four AMs with different numbers of states (in the range from 2447 to 11489).

Next, we trained AMs using the proposed triphone mapping as described in Sec. 2. In order to achieve the same range of trained states, we set the limit of minimum occupancy $N$ of triphones in the range from 200 to 2850.

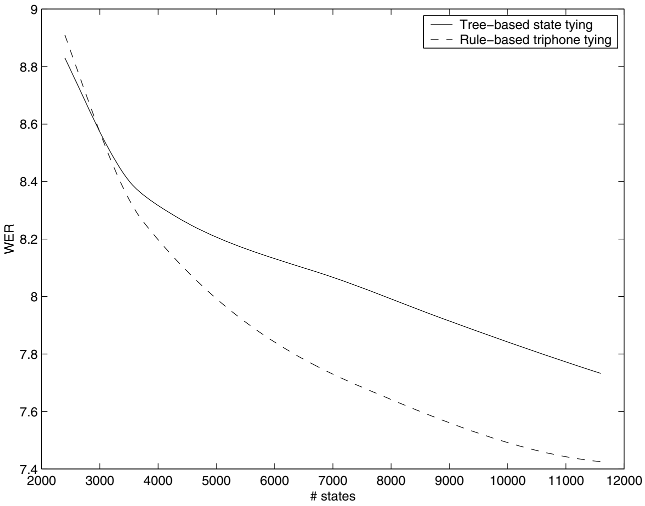


**Fig. 3.** Tree-based state tying compared to triphone tying on the number of trained HMM states. Four acoustic models were trained for each function with 2450, 3700, 7450 and 12000 states. The results were then interpolated to get smooth functions.

Fig. 3 shows the results of tree-based state tying and triphone mapping. For small acoustic modeling up to 3000 states, tree-based state tying slightly outperforms triphone mapping (e.g. for models with 2500 states). For bigger models, at the size typical for large vocabulary continuous speech recognition (LVCSR) training (more than 3000 states), rule-based triphone mapping achieves better word error rate (WER). Relative decrease of word error rate was 4.23% for models with 7500 states, and 4.13% with 11500 states.

## 4 Discussion

We showed that the rule-based triphone mapped HMM system achieves better WER for models typical for LVCSR training. This result poses an interesting question: Why does the triphone mapped HMM system, the model tying approach, perform better than the state tying approach, if it was already shown that state tying systems consistently out-perform model clustered systems (see e.g. [2])?

In the process of building the triphone mapped HMM system we emphasized, that while the standard state tying system clusters the states from the single Gaussian models (due to performance reasons), trained roughly in $1/3$ from all the training time, the triphone mapped system can cluster the models from the multiple Gaussian Mixture Models (GMMs), trained roughly in $4/5$ from all the training time. Both tying systems work with single Gaussian models for the calculation of distance metric; however, the triphone mapping can be easily applied later in the training process, when monophone models are much better trained using multiple Gaussians. In order to verify this hypothesis, we forced the process of building the triphone mapped HMM system to be more similar to the building the state tying system (cloning and clustering the triphones from the single Gaussian models):

1. Training of monophones models with single Gaussian mixtures
2. The state output distributions of the monophones are cloned, triphone mapping is applied
3. The triphone tied system is trained
4. The number of mixture components in each state is incremented and the models are trained again

We trained the triphone mapped HMM system using this modified process above, and for 12000 states we obtained similar performance as with the state tying system and the same model size. We can thus conclude that the main gain in performance is due to latent application of triphone mapping. Having well trained monophones using multiple Gaussians distributions, the cloned triphones are better initialized than with single Gaussians monophones. The performance change at 3000 states is probably related to the amount of training data available. The more data we have, the more states we can robustly train.

The process of triphone mapping is language independent, and can be further tuned with an application of different weights for the left and right contexts. Data-driven triphone mapping belongs to our future work as well.

**Acknowledgements.** This work was supported in part by the EU grant (ITMS 26240220064).

## References

1. Bahl, L.R., de Souza, P.V., Gopalakrishnan, P.S., Nahamoo, D., Picheny, M.A.: Decision trees for phonological rules in continuous speech. In: ICASSP 1991, pp. 185–188. IEEE Computer Society, Washington, DC, USA (1991)
2. Young, S.J., Odell, J.J., Woodland, P.C.: Tree-based state tying for high accuracy acoustic modelling. In: Proceedings of the workshop on Human Language Technology. HLT 1994, pp. 307–312. ACL, Stroudsburg (1994)
3. Baker, J., Deng, L., Khudanpur, S., Lee, C.H., Glass, J., Morgan, N., O'Shaughnessy, D.: Updated MINDS report on speech recognition and understanding, Part 2 (DSP Education). IEEE Signal Processing Magazine 26(4), 78–85 (2009)
4. Prince, A., Smolensky, P.: Optimality Theory: Constraint Interaction in Generative Grammar. Blackwell, Oxford (1993/2004)
5. Young, S., Woodland, P.C.: State clustering in hidden Markov model-based continuous speech recognition. Computer Speech & Language 8(4), 369–383 (1994)
6. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Ovell, J., Ollason, D., Valtchev, D.P.V., Woodland, P.: The HTK Book (for v3.4.1), Cambridge (2009)
7. Lee, A., Kawahara, T., Shikano, K.: Julius – an Open Source Real-Time Large Vocabulary Recognition Engine. In: Proc. of the European Conference on Speech Communications and Technology (EUROSPEECH), Aalborg, Denmark (September 2001)

8. Johansen, F.T., Warakagoda, N., Lindberg, B., Lehtinen, G., Kačič, Z., Žgank, A., Elenius, K., Salvi, G.: The COST 249 SpeechDat multilingual reference recogniser. In: Proc. of the 2nd Intl. Conf. on LREC, Athens (May 2000)
9. Barras, C., Geoffrois, E., Wu, Z., Liberman, M.: Transcriber: development and use of a tool for assisting speech corpora production. Speech Communication 33(1–2) (January 2000)
10. Staš, J., Hládek, D., Juhár, J.: Language Model Adaptation for Slovak LVCSR. In: Proc. of the Intl. Conference on AEI, Venice, Italy, pp. 101–106 (2010)

# Semantic Relatedness for Named Entity Disambiguation Using a Small Wikipedia*

Izaskun Fernandez[1], Iñaki Alegria[2], and Nerea Ezeiza[2]

[1] Tekniker-IK4
ifernandez@tekniker.es
http://www.tekniker.es/
[2] IXA Group
{i.alegria,n.ezeiza}@ehu.es
http://ixa.si.ehu.es/Ixa

**Abstract.** Resolving Named Entity Disambiguation task with a small knowledge base makes the task more challenging. Concretely, we present an evaluation of the state-of-the-art methods in this task for Basque NE disambiguation based on the Basque Wikipedia. We have used MFS, VSM, ESA and UKB for linking any ambiguous surface NE form occurrence in a text with its corresponding Wikipedia entry in the Basque Wikipedia version. We have analysed their performance with different corpora and as it was expected, most of them perform worse than when using big Wikipedias such as the English version, but we think these results are more realistic for less-resourced languages. We propose a new normalization factor for ESA to minimise the effect of the knowledge base size.

**Keywords:** Named Entity Disambiguation, Semantic Relatedness, Wikipedia.

## 1 Introduction

Named Entity Disambiguation (NED) is the task of exploring which real person, place, event... is referred to by a certain surface form. For example, NED must decide whether, in a given context, the surface form *Amstrong* refers to either the cyclist *Lance Armstrong* or the astronaut *Neil Armstrong*.

In many respects, named entities disambiguation task is similar to word sense disambiguation (WSD). WSD task, which has supported large-scale evaluations (SENSEVAL editions[1]), aims to assign dictionary meanings to all the instances of a predetermined set of polysemous words in a corpus. However, these evaluations do not include proper name disambiguation and omit named entity meanings from the targeted semantic labels and the development and test contexts.

Recently, a couple of evaluation campaigns with a specific task dealing with NED have been defined within the TAC-KBP track (Knowledge Base Population), organized in 2009 [9] and 2010 [7] respectively. The entity linking task consists in determining,

[*] This research was partially supported by MICINN (TIN2010-20218) and the Basque Government (EMAITEK2011:Medicina Digital).

[1] http://www.senseval.org/

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 276–283, 2011.

for a given NE surface form and a particular context, which of the entries in a knowledge base (KB) the NE refers to, provided there is one. In the TAC-KBP track, the English Wikipedia was the KB.

But those results have not been tested yet for smaller Wikipedia versions, such as those written in less-resourced languages. In this work we analyse how characteristics of a smaller Wikipedia (the Basque Wikipedia) change compared to a larger one (the English Wikipedia) and how they can affect the performance of named entity disambiguation systems. The distinctive features of a smaller Wikipedia are the following:

- *Less ambiguity*: since there are fewer articles, the list of possible candidates for an ambiguous entity in the KB will be shorter.
- *More NE without an entry in the KB*: (related to the previous feature) it is possible that more named entities will not be represented in the Wikipedia. This characteristic depends on the KB, but also on the target corpus.
- *Shorter articles*: the percentage of short articles is higher, so the amount of information for disambiguation is smaller.
- *Less links*: (related to the previous feature) we can presume that less links will be available to extract new information.

Although it seems that the first feature leads to an easier task, the rest of them make the task more challenging when working with a small KB.

Our aim is to test some of the most popular NED methods described in the literature for the task of linking named entities in texts to Wikipedia articles, using a small Wikipedia (the Basque Wikipedia, eu.wikipedia.org) for this purpose. Concretely we test bag-of-words in a vector space model, the ESA algorithm and a graph-based model (UKB) in this context.

The remainder of the paper is organized as follows. Section 2 describes the state of the art in NED task. Section 3 describes the disambiguation strategy we have applied, specifying the resources and the methods used for it. Section 4 describes the evaluation methodology and in the Section 5 we present the results. Finally, we show some conclusions and future work.

## 2 Related Work

In the last few years growing interest has been shown in the task of linking Wikipedia and NED. In [3] the authors use several characteristics from an English Wikipedia dump, such as the text of the entries, categories, redirection pages and hyperlinks, which are used to train a supervised model (based on SVM) to link an entity occurrence to a Wikipedia page. They evaluate the system for person name disambiguation, reporting accuracies of 55.4% to 84.8%.

In [4], Cucerzan formalized the disambiguation paradigm, which is based on Wikipedia and includes similar information as the one described in the previous paper in a vector-space model. More specifically, the vectorial representation of a document is compared with the vectorial representation of the Wikipedia entities, which are represented as an extended vector with two main components, corresponding to context and category information. The reported accuracy is 88% to 91%.

Gabrilovich and Markovitch [5] presented Wikipedia-based ESA (Explicit Semantic Analysis) to compute semantic relatedness of documents. ESA works by first building an inverted index from words to all Wikipedia articles that contain them. Then, it estimates a relatedness score for any two documents by using the inverted index to build a vector over Wikipedia articles for each document and by computing the cosine similarity between the two vectors.

Graph-based models have been also used to face this problem [6]. In this work, the authors propose a method, which uses a graph model using multiple features extracted from Wikipedia, to estimate Semantic Relatedness over the Wikipedia-based graph. They exploit the obtained relatedness values to resolve the NED problem, obtaining 91.46% and 89.83% accuracy in two different evaluation sets.

A wide range of methods and combinations have been developed for the entity linking task in TAC2009 and TAC2010 ([9], [7]). This task is similar to the one we explain in this paper: given a name and its context, the system must decide whether this name corresponds to an entry in a database from Wikipedia and, if so, which one.

## 3 Experimental Settings

The NED process aims to create a mapping between the surface form of an entity and its unique meaning in the KB if it exists. So a dictionary indicating all possible entries in the KB for a surface form is needed. In this work we use a Basque Wikipedia dump dated March 2006 as KB so as to build the mapping dictionary and extract features for the different disambiguation methods. The algorithms, given a surface form of a named entity and its occurrence context (a paragraph), must decide whether the name corresponds to an entry in the Wikipedia and, if so, to which one. For the evaluation, we have used a news corpus. All these resources are detailed in the following sections.

### 3.1 Resources

**Named Entity Ambiguity Dictionary.** To build the mapping dictionary based on the Basque Wikipedia, we try to derive all the possible ambiguous forms for each Wikipedia entry. Once all the surface forms are generated, we represent them in a dictionary where each surface form is defined by a set of Basque Wikipedia entries. We apply the following strategy to generate the dictionary:

- *The title itself* is considered a possible surface form for the entry.
- When a title has more than one word, we *generate all the possible combinations replacing each word (except the last one) with its initial*, and we add them as surface forms. For instance, to deal with the entry *Juan Jose Ibarretxe*, we add the surface forms *Juan J. Ibarretxe*, *J. Jose Ibarretxe* and *J. J. Ibarretxe*.
- *Each word in the title* is considered a surface form. For instance, three new entries are created in the previous example: *Juan*, *Jose* and *Ibarretxe*.

While exploiting the Wikipedia characteristics we also enrich the set of surface forms. Since if an entry in Wikipedia refers to another entry, it should be linked to its main entry, those anchors can also be considered surface forms for the target entry. As redirect entries represent just another form to mention the linked entry, they are also added to the set of surface forms for the target entry.

Finally, we also use the content of disambiguation pages. Unlike in redirect pages, disambiguation titles are considered surface forms for the entries listed in them, because, as the name implies, they are entries that refer to different Wikipedia entries, so they are ambiguous names.

**News Corpus.** Since there is no standard Basque corpus defined for the NED evaluation task, we have generated a repository for that purpose using pieces of news of the 2002 year edition of the *Euskaldunon Egunkaria* newspaper. To be precise, we have used an annotated version of this news corpus, processed in the context of HERMES project (http://nlp.uned.es/hermes/). The corpus has 40,648 articles and 135,505 NEs.

It is very common in news texts to use the entire or long form of an entity (i.e. *Aimar Olaizola*) in its first occurrence and shorter forms (i.e. *Aimar* or *Olaizola*) in the later occurrences within the same item of news. These short surface forms have higher ambiguity, since they are not as specific as the longer ones, so they can refer to a larger set of NEs.

We do not take advantage of this feature during the disambiguation task, because the aim of our work is to evaluate different methods for short contexts which might not have more than one instance of the same entity. However, it is very useful to generate a corpus that does not need any hand revision for the evaluation. We have created a test-corpus (Corpus A) which only includes texts that meet the following conditions: they have instances of short entities and their longer unambiguous forms in the same item of news; and the unambiguous forms have their corresponding Wikipedia entry.

This way, for every ambiguous NE, we know which Basque Wikipedia entry it must be linked to as a result of the disambiguation. Looking at the previous example, if *Aimar Olaizola* appears in Wikipedia and it is unambiguous at the piece of news we are analysing, the paragraphs where *Aimar* or *Olaizola* surface forms occur will be selected for this test corpus.

Another test-corpus was built in a regular way, collecting news paragraphs with at least one entity, no matter if there was a longer NE containing it along the news or if it was represented in the Wikipedia. Since there was no automatic way to know which the corresponding Wikipedia entry was, if any, for each NE in this example set, the corpus (henceforth Corpus B) was manually disambiguated, linking each NE occurrence to its corresponding Wikipedia entry, when possible. Even if in this work no training process has been applied, Corpus B was divided into two groups in order to use one for the tuning process (Corpus B-dev) and the second one for evaluation (Corpus B-eval).

**Table 1.** Evaluation corpora

| | **# Examples** | **# Ambiguous Ex.** | **NIL** | **Ambiguity** |
|---|---|---|---|---|
| **Corpus A** | **6,500** | **4,376** | **0** | **67.32%** |
| **Corpus B-dev** | **532** | **295** | **70** | **55.45%** |
| **Corpus B-eval** | **500** | **300** | **63** | **60%** |

Table 1 summarizes the main features of the mentioned corpora, including the number of NE to disambiguate; the number of the ambiguous examples in the set, considering non-ambiguous those surface forms that are only defined by one Wikipedia entry in the NE ambiguity dictionary; the number of examples that have no corresponding NE disambiguation form represented in Wikipedia, in which case the system should answer NIL; and finally the ambiguity rate.

### 3.2 Methods

The methods we want to test with a small Wikipedia KB are some of the most popular in the literature: bag-of-words in a vector space model (VSM), ESA and UKB, a graph-based model. As it is usual in WSD, the baseline is calculated using the most frequent entity among the candidates (MFS). To compute the most frequent entity, we use the number of in-links of each Wikipedia entry. The entry with the highest number of in-links is considered the most frequent one.

**VSM.** Vector Space Model [2] for the resolution of NE ambiguity represents all the Wikipedia entries using bag-of-words vectors, being each word position measured with its *tfidf* value in the corresponding Wikipedia entry. Based on this information, when a new NE occurrence has to be disambiguated, its context is represented in the same vector space with bag-of-words modelling. This vector is compared to the ones corresponding to the Wikipedia disambiguation pages for the given ambiguous NE form defined at the dictionary, computing the cosine of each vector pair. The Wikipedia entry with the highest cosine value will be the one proposed as the disambiguation form.

**ESA.** Explicit Semantic Analysis (ESA) [5] is a vector space comparison algorithm based on Wikipedia articles. For a candidate text, each dimension in its ESA vector corresponds to a Wikipedia article, with the score being the similarity of the text with the article text, subject to *tfidf* weighting. The relatedness of two texts is computed as the cosine similarity of their ESA vectors.

Since in NED task the aim is to compare an input text with a set of Wikipedia entries, it is not necessary to construct the entire ESA vector. Estimating the similarity measures for that set of Wikipedia entries is enough, as it is formalized by Sorg and Cimiano [10]. So what we need is just to compute the similarity values for a given ESA vector dimension.

For computing the association strength between an $A_j$ Wikipedia article and a T input text, ESA applies the following metric:

$$ESASimilarity(T, A_j) = \sum_{w_i \epsilon T} v_i * k_j$$

*where* $k_j$ *is the tfidf value of* $w_i$ *in* $A_j$*, and* $v_i$ *the tfidf value of* $w_i$ *in T*

**Balanced ESA.** The association strength defined for ESA tends to promote short articles over long ones when articles in the KB are very different in terms of length. This effect disappear when articles are similar in extent or at least of a minimum length, as it happens when this algorithm is tested in a bigger KB such as the English Wikipedia. In order to reduce this adverse effect, we have introduced a normalization factor to the original ESA strength association measure, which takes into account the number of words shared between the Wikipedia article and the input text. So the new association strength estimation is defined as follows:

$$bESAsimilarity(T, A_j) = ESASimilarity(T, A_j) * \frac{Count_T}{|T|}$$

*where* $Count_T$ *is the number of words shared by a T input text and an* $A_j$ *article*

**UKB.** UKB [1] is a Personalized PageRank method that aims to obtain a relatedness score between a pair of texts by performing random walks over a graph to compute a stationary distribution for each text. In order to apply it in the context of NED based on the Basque Wikipedia, it is necessary to build two resources: the Basque Wikipedia as a graph, and a dictionary.

In order to construct the graph structure of the Basque Wikipedia 2006 dump, we simply treat the articles as vertices, and the links between articles as edges as in [12]. The graph has 63,106 vertices and 458,026 edges.

The aim of this task is to disambiguate surface forms linking them with a particular Wikipedia article, provided there is one. Therefore, the mapping dictionary should have available correspondences between NE ambiguous forms, surface forms, and the set of Basque Wikipedia articles that could be their disambiguation forms. This resource has been described in 3.1.

## 4 Evaluation and Results

The goal of the NED tool is to give, for a certain NE, the Wikipedia entry with the highest score among the candidates. When no candidate is found in the dictionary, no answer is possible and the system must return NIL. When there is a scoreless tie, the tool can have one of the following behaviours:

1. Silent mode. The system does not take any decision on a tie.
2. Tie-break mode. The system makes a decision by applying a random answer or MFS.

In silent mode, we evaluate the performance of the algorithms in terms of *F-Score*. As there is neither special treatment nor a defined thershold for NIL prediction, the system only returns NIL when we force an answer (in tie-break mode) and there is no candidate. Otherwise, it decides at random or using MFS. Thus, in tie-break mode, we treat NIL as a possible answer and, we compute *accuracy* accordingly.

Table 2 shows the results obtained by each algorithm. The first column describes the results of the baseline system (MFS), which is used as a reference in tie-break mode. In silent mode, we want to point out that, unexpectedly, ESA obtains the worst results, because of the low recall. We think that the small size of the Wikipedia articles makes it difficult to obtain strong similarity measures when we compare them with the context paragraphs of the target NE instance. However, we intend to take a deeper view on the results to confirm this suspition. Using bESA we obtain better results than with ESA or VSM. Nevertheless, UKB obtains the best results, outperforming significantly the rest

**Table 2.** Results of the algorithms

| | MFS | VSM | ESA | bESA | UKB |
|---|---|---|---|---|---|
| **A – Silent** | 68.32% | 70.15% | 66% | 71.25% | 81.91% |
| **A – Random** | 68.32% | 74.03% | 71% | 75.94% | 82.8% |
| **A – MFS** | 68.32% | 75.53% | 72.43% | 77.66% | 82.8% |
| **B-Dev - Silent** | 72% | 69% | 59.4% | 66.6% | 75.7% |
| **B-Dev - Random** | 72% | 70.3% | 60.9% | 67.8% | 75.9% |
| **B-Dev - MFS** | 72% | 70.4% | 61% | 68% | 75.9% |
| **B-Eval - Silent** | 70.4% | 67% | 58% | 65% | 76% |
| **B-Eval - Random** | 70.4% | 69% | 61.2% | 68.4% | 76.2% |
| **B-Eval - MFS** | 70.4% | 70% | 61.6% | 68.4% | 76.2% |

of the methods and it is the only method that achieves better results than MFS in Corpus B.

Corpus B-Dev and Corpus B-Eval have similar size (500 NEs) and the results are comparable between them, but there is a drop of 5-7 points with respect to Corpus A in the case of ESA, bESA and UKB. We think this is due to the low recall and the bad assignment of the NIL choice, but it requires deeper analysis.

In tie-break mode, as it was expected, the results are better when MFS is applied. For Corpus B we observe that the results are 3 points better in average, while for Corpus A the difference is higher, because the NEs in Corpus A have always an answer in Wikipedia. The improvement is not so important for UKB. The main reason is that UKB has significantly higher recall, so there are less scoreless ties to break.

## 5 Conclusions

We have presented the work we have done in the field of Named Entity Disambiguation (NED) based on the Basque Wikipedia. Being the Basque Wikipedia a small KB, we have tested most of the state-of-the-art algorithms in order to evaluate their performance using small resources instead of big KBs.

Despite being ESA the most popular algorithm for semantic relatedness estimation, we have seen that for a small Wikipedia, with short and few entries, it does not perform so well. To minimise the negative effect caused by the KB size, we have proposed a new normalization factor for ESA, which provides better performance than the original, even getting one of the best performances in terms of accuracy and stability.

The UKB algorithm has not achieved very good results for NED task using the English Wikipedia as KB for graph construction (A. Soroa, pers. comm.). But surprisingly, for Basque, not only does it perform well, but it has also turned out to be the best of the tested algorithms for every evaluation corpora. We are examining the results on Corpus B-Dev in order to clarify the reasons of this.

We think that improving the NIL assignment will be a key work for the future. Finally, we consider interesting as future work to combine the different algorithms to get better results, especially in terms of recall.

## References

1. Agirre, E., Soroa, A.: Personalizing PageRank for Word Sense Disambiguation. In: Proceedings of the 12th Conference of the European chapter of the Association for Computational Linguistics, pp. 33–41 (2009)
2. Baeza, R., Ribeiro, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
3. Bunescu, R.C., Pasca, M.: Using Encyclopedic Knowledge for Named entity Disambiguation. In: 1th Conference of the European Chapter of the Association for Computational Linguistics, pp. 9–16 (2006)
4. Cucerzan, S.: Large-Scale Named Entity Disambiguation Based on Wikipedia Data. Empirical Methods in Natural Language Processing (2007)
5. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 6–12 (2007)
6. Gentile, A.L., Zhang, Z., Xia, L., Iria, J.: Cultural Knowledge for Named Entity Disambiguation: A Graph-Based Semantic Relatedness Approach. Serdica Journal of Computing 4(2), 217–242 (2010)
7. Ji, H., Grishman, R., Dang, H.T., Griffitt, K., Ellis, J.: Overview of the TAC 2010 Knowledge Base Population Track. In: Proceedings of Text Analysis Conference (2010)
8. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for wordsense identification. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 265–283. MIT Press, Cambridge (1998)
9. McNamee, P., Dang, H.T.: Overview of the TAC 2009 Knowledge Base Population track. In: Proceedings of the Second Text Analysis Conference (2009)
10. Sorg, P., Cimiano, P., Enriching, P.: the crosslingual link structure of Wikipedia-A classification-based approach. In: Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (2008)
11. Strube, M., Ponzeto, S.P.: WikiRelate! Computing semantic relatedness using Wikipedia. In: Proceedings of the AAAI 2006, pp. 1419–1424 (2006)
12. Yeh, E., Ramage, D., Manning, C., Agirre, E., Soroa, A.: WikiWalk: Random walks on Wikipedia for Semantic Relatedness. In: Proceedings of ACL Workshop TextGraphs-4: Graph-based Methods for NLP (2009)

# Speaker-Clustered Acoustic Models Evaluated on GPU for On-line Subtitling of Parliament Meetings

Josef V. Psutka, Jan Vaněk, and Josef Psutka

Department of Cybernetics, West Bohemia University, Pilsen, Czech Republic
{psutka_j,vanekyj,psutka}@kky.zcu.cz,
http://www.kky.zcu.cz

**Abstract.** This paper describes the effort with building speaker-clustered acoustic models as a part of the real-time LVCSR system that is used more than one year by the Czech TV for automatic subtitling of parliament meetings broadcasted on the channel ČT24. Speaker-clustered acoustic models are more acoustically homogeneous and therefore give better recognition performance than single gender-independent model or even gender-dependent models. Frequent changes of speakers and a direct connection of the LVCSR system to the audio channel require an automatic switching/fusion of models as quickly as possible. An important part of the solution is real time likelihood evaluations of all clustered acoustic models, taking advantage of a fast GPU(Graphic Processing Unit). The proposed method achieved a WER reduction to the baseline gender-independent model over 2.34% relatively with more than 2M Gaussian mixtures evaluated in real-time.

## 1 Introduction

Recently, we introduced the system for automatic subtitling of the 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], [2] and [3]) for other application see [4]. An unpleasant problem that accompanies the automatic speech recognition of deputies is frequent and sometimes very rapid changes of speakers. It complains about the use of the on-line speaker adaptation techniques, which need relatively a longer part of speech for adaptation. To avoid using common speaker adaptation techniques we suggested a method, which operates simultaneously with several speaker clustered acoustic models. The fast model switching/fusion makes possible to "adapt" the ASR system on-line to a new voice within a few frames. An increase in computational demands during calculations of output likelihoods was eliminated by using the GPU(Graphic Processing Unit).

## 2 Methods

### 2.1 Unsupervised Clustering

Recently, we describe [5] an automatic clustering algorithm that divides speakers by voice into homogeneous classes. It is based on iterative k-means-like approach composed from three main steps. The algorithm starts with partitioning of initial training

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 284–290, 2011.

data into a predefined number of clusters (randomly or using any relevant prior information). In the first step, the acoustic models for the individual clusters are trained or adapted. The second step consists of a criterion calculation for all training data across all models. The last step of the iteration is the data (uttered sentences in our case) reassignment to the cluster with the best criterion value. If the percentage of cross-assigned utterances in all the clusters decreases bellow some predefined value, the iteration process is stopped.

For more than two clusters, the hierarchical or direct variant of the algorithm can be applied. In the hierarchical case, the initial data are split into two clusters and the actual cluster with the largest criterion is than split into another two clusters. This process continues until the target number of clusters is achieved. The main advantage of this approach is the speed. Because only a small part of the data is processed in after-initial splitting steps. The disadvantage of this approach is a possibility to create ”gaps” between the binary-tree branches. In contrast, the direct approach splits the whole training set directly into target number of clusters. It is very computationally intensive. The computational demands grow linearly per iteration with the number of clusters. Moreover, the required number of iterations usually grows as well. However, the quality of the models set produced by direct approach is slightly higher. Disadvantage of the direct approach next to the computational intensity is a higher sensitivity to the initial data partitioning.

Various criteria that are used for acoustic model training can be also used for the clustering. It could be the traditional Maximal Likelihood criterion (ML) but also discriminative criteria, e.g. Maximum Mutual Information (MMI) or Minimum Phone Error (MPE). After obtaining the training data from the clustering algorithm (list of sentences in our case), the final models set has to be trained. Appropriate training techniques were examined in [6] and [5]. The best performance was achieved using the discriminative criterion. If the final number of clusters is small and the clusters are large enough, the full discriminative training procedure is appropriate. In the case of the higher number of clusters with some relatively empty clusters, the discriminative adaptation techniques are helpful. The models do not differ so much but their parameters are estimated robustly.

### 2.2 Acoustic Models Fusion

Recently, various techniques for acoustic models switching/fusion were proposed (see [3] for details). All presented techniques were designed for the real-time applications therefore only a small history for actual processed frames is needed. Results of an extensive experimental work suggest that the best solution is a weighted sum with exponential forgetting. This method can be written in the form of

$$\hat{P}(s_i|\mathbf{o}_t) = \sum_{k=1}^{M} w_t^k P_k(s_i|\mathbf{o}_t). \qquad (1)$$

where $P_k(s_i|\mathbf{o}_t)$ is an output probability of the state $s_i$ of the $k$-th acoustic model, $\hat{P}(s_i|\mathbf{o}_t)$ is the new evaluated state's probability and $M$ is number of acoustic models. The weights in the time $t$ are computed as

$$w_t^k = \frac{\alpha P_{t-1}(\lambda_k) + (1-\alpha)P(\lambda_k|\mathbf{o}_t)}{\sum_{l=1}^{M} \alpha P_{t-1}(\lambda_l) + (1-\alpha)P(\lambda_l|\mathbf{o}_t)}. \tag{2}$$

where $\alpha$ parameter is set to 0.95 and where $P_0(\lambda_k)$ are set to zero for all acoustic models $k$.

### 2.3 GPU Accelerated Acoustic Model Evaluation

The computation of acoustic model likelihoods accounts for the largest processing part in automatic speech recognition systems. We use GPU cards to offload this computation-intensive part from CPU. Our optimized algorithm efficiently exploits the GPU [8]. The efficiency together with the high GPU performance enable us to evaluate very large acoustic models much faster than in real-time. Evaluation of several acoustic models together is now possible even on low-end or laptop GPUs.

Our implementation splits the acoustic model evaluation into the data-parallel blocks. The individual blocks evaluate 8 or 16 feature-vectors together with 64 tied-states which are based on Gaussian mixture models with diagonal covariance matrix. The number of evaluated feature-vectors is a trade-off between the recognition system delay and the system performance. In the case of a small number of feature vectors, the overhead of CPU-GPU communication together with GPU-kernel management is significant. Fully asynchronous GPU handling is used for the maximum total system performance. Two likelihood-buffers are prepared. One is used by the decoder and the other is asynchronously filled by the results of evaluation of the next feature-vector window. In the most cases, the GPU evaluation is faster than decoder part. Therefore, CPU-implemented decoder is a bottleneck of the entire system and total recognition speed depends on the decoder performance.

## 3 Train Data Description

### 3.1 Annotated Data

The training corpus consists of two different parts. The first one contains 100 hours of parliament speech records collected till 2010. All this data has been manually annotated and carefully revised (see details in [9], [10]). However, after the parliament election in 2010, more than 50% of the parliament members were changed.

### 3.2 Unsupervised Data

The second part of the corpus contains 300 hours of speech. This huge part of parliamentary speeches was collected from deputies elected in 2010. There were no manual transcriptions available for this data. But the shorthand records of all Parliament meetings must be (by law) available for public use on the Internet. Unfortunately, these shorthand records are amended to avoid slips of the tongue and to meet the grammatical rules, so they do not meet the demands for exact transcriptions suitable for acoustic model training. Anyway, we can use these transcriptions of the individual meetings to create the meeting-specific language models by combination of the language model trained from the meeting transcriptions (dynamic LM) with the standard language model (static LM). Static language model was trained on about 27M tokens of the normalized Czech Parliament meeting transcriptions (Chamber of Deputies only) from different electoral periods. Dynamic language models were trained on meeting transcriptions containing from 3k to 100K tokens. Resulted trigram language model with modified Kneser-Ney smoothing was trained by SRI Language Modeling Toolkit [11]. This approach reduced the word error rate of the recognized transcriptions to about 50 %.

Since the recognition process was not error-free, some technique for confidence tagging of the recognized words was used to choose only well-recognized segments of the speech that were taken into the acoustic model training process. We used the posterior word probabilities computed on the word graph as a confidence measure [12]. To use only the trustworthy segments of the speech, we applied a quite strict criterion for the word selection - only the words, which had confidence greater than 0.99 and their neighboring words with the same confidence (greater than 0.99), were selected. This ensures that the word boundaries of selected words are correctly assigned for retraining of acoustic model.

# 4 Experimental Setup

## 4.1 Acoustic Processing

The digitization of an analogue signal was provided at 44.1 kHz sample rate and 16-bit resolution format. The front-end worked with PLP parameterization with 27 filters and 12 PLP cepstral coefficients with both delta and delta-delta sub-features (see [13] for methodology). Therefore one feature vector contains 36 coefficients. Feature vectors are computed each 10 milliseconds (100 frames per second). Cepstral mean normalization (CMN) was used in order to reduce the effect of constant channel characteristics.

## 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 the Czech triphones is large, phonetic decision trees were used to tie the states of the 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 48 mixtures of multivariate Gaussians for each of 5385 states. A silence model was trained by borrowing the most relevant Gaussians from all non-speech HMMs in proportion to their state and mixture occupancies. Thus the resulting silence model contained 253 mixtures on average per state. The prime 48 Gaussians triphone acoustic model trained by Maximum Likelihood (ML) criterion was made using HTK-Toolkit v.3.4 [14]. At second, final models were obtained via two iterations of MMI-FD discriminative training [6] or [7].

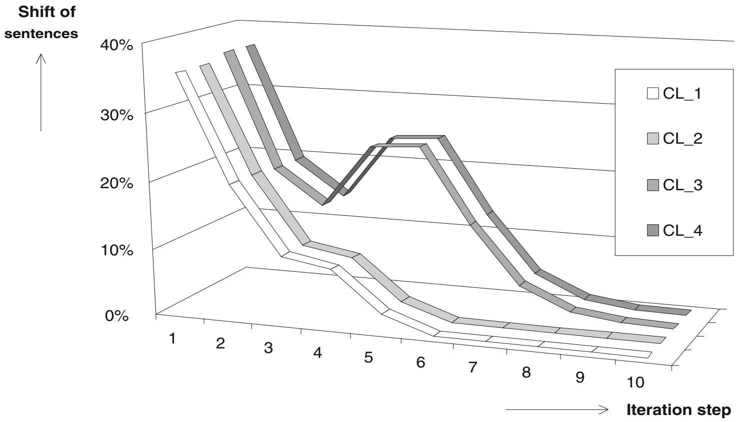


**Fig. 1.** An example of behaviour of stopping criterion $(4Cl_h)$

### 4.3 Unsupervised Speaker Clustering

As was presented in [3], by splitting via manual male/female markers and creating the gender-based acoustic models we achieved a significant gain in terms of the recognition accuracy than a simple speaker-independent acoustic model. This method is the most popular method how to split training data into two more acoustically homogeneous classes [16], [15]. As can be seen in [5] the increasing number of the speaker-clusters brings even better recognition results in comparison with the gender-dependent acoustic models. The whole training corpus was split hierarchically into two, four and eight acoustically homogeneous classes via the algorithm introduced in the Subsection 2.1. However, in all cases, the initial splitting was achieved randomly because additional speaker/sentence information was unavailable (especially for the second part of the training corpus). The stopping criterion in all presented experiments was based on the shift between clusters (sentences, which were moved from the one cluster to another in the consecutive steps) less than 1% of sentences. An expample of such criterion can be seen on the Figure 1.

### 4.4 Tests Description

The test set consists of 1 hour of the special part of the parliament meetings - interpellation. Interpellation is the right of a parliament to submit questions (oral or formal) to the government. This part of the parliamentary speeches was chosen because of the limited time for one speaker (2 minutes per question). This portion of speech contains 15 different speakers (11 male and 4 female) since each speaker has the right to submit several questions. All recognition experiments were performed with a trigram language models with modified Kneser-Ney smoothing that was trained by SRI Language Modeling Toolkit [11]. The language model was trained on about 27M tokens of normalized Czech parliament transcriptions. The model contained 192k words with OOV amounting 4%. The perplexity of the recognition task was 547.

## 5 Results

In all our experiments, the word error rate (WER) as well as clustering criterion (Maximal Likelihood criterion and Maximum Mutual Information) were evaluated. We tried only the hierarchical division methods to split all the training sentences (175k) into two, four, or eight clusters. This type of division method was used according to former experiments (see [5] for details).

The hierarchical division method means that we divided the training set into two classes ($2Cl_h$) and than each class was split again into another two classes ($4Cl_h$) and so on (finally we had eight clusters $8Cl_h$). The recognition results as well as some parameters which describe the clustering criterion are shown in the Table 1.

**Table 1.** Recognition results

| | WER [%] | | | |
|---|---|---|---|---|
| | Ideal | Real | ML Criterion | MMI |
| baseline | 13.70 | | - | - |
| $2Cl_h$ | 13.18 | 13.53 | 68.49 | -3.2860 |
| $4Cl_h$ | 12.86 | 13.47 | 68.68 | -3.2231 |
| $8Cl_h$ | 12.60 | 13.38 | 68.83 | -3.1809 |

The column *Ideal WER* shows the recognition results for the ideal off-line recognition, where the tests were performed on the list of single speaker sentences across all clustered acoustic models. On the other hand the column *Real WER* shows the recognition results for the real-time recognition, where the fusion of all clustered acoustic models was applied. From the obtained results we can see that the best recognition result was achieved for eight clusters ($8Cl_h$). The number of on-line evaluated Gaussians in this case is more than 2M.

## 6 Conclusion

The goal of this paper was to describe our work with building speaker-clustered acoustic models in a real-time LVCSR system for automatic subtitling of Parliament meetings that are broadcasted on the TV channel ČT24. To be able to use the speaker-clustered acoustic models in the task of on-line recognition of parliamentary speeches with the frequent changes of speakers, we suggested the fast switching/fusing method of acoustic models evaluation. This approach works with the support of a GPU unit and is able to enumerate 2M Gaussian mixtures in real-time. The proposed method achieved a WER reduction by more than 2% relatively compared to the baseline gender-indendent model.

**Acknowledgments.** This research was supported by the Technology Agency of the Czech Republic, project No. TA01011264.

## References

1. Pražák, A., et al.: 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. Pražák, A., Müller, L., Psutka, J.V., Psutka, J.: LIVE TV SUBTITLING - Fast 2-pass LVCSR System for Online Subtitling. In: SIGMAP 2007: Proceedings of the Second International Conference on Signal Processing and Multimedia Applications, pp. 139–142. INSTICC Press, Lisbon (2007)
3. Vaněk, J., Psutka, J.V.: Gender-dependent acoustic models fusion developed for automatic subtitling of Parliament meetings broadcasted by the Czech TV. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 431–438. Springer, Heidelberg (2010)
4. Neto, J., et al.: Broadcast News Subtitling System In Portuguese. In: Proceedings of the ICASSP, Las Vegas, USA (2008)
5. Vaněk, J., Psutka, J.V., Zelinka, J., Pražák, A., Psutka, J.: Training of Speaker-Clustered Acoustic Models for Use in Real-Time Recognizers. In: SIGMAP 2007: Proceedings of the Second International Conference on Signal Processing and Multimedia Applications, pp. 131–135. INSTICC Press, Lisbon (2009)
6. 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, vol. 5729, pp. 331–338. Springer, Heidelberg (2009)
7. Povey, D., Woodland, P.C.: Improved discriminative training techniques for large vocabulary continuous speech recognition. In: IEEE International Conference on Acoustics Speech and Signal Processing, Salt Lake City, Utah (2001)
8. Vaněk, J., et al.: Acoustic Likelihoods Computation Optimized for NVIDIA and ATI/AMD Graphics Processors. Submited to IEEE Signal Processing Magazine (2011)
9. Radová, V., Psutka, J.: Recording and Annotation of the Czech Speech Corpus. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2000. LNCS (LNAI), vol. 1902, pp. 319–323. Springer, Heidelberg (2000)
10. Kolář, J., Švec, J.: The Czech Broadcast Conversation Corpus. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 101–108. Springer, Heidelberg (2009)
11. Stolcke, A.: SRILM - An Extensible Language Modeling Toolkit. In: International Conference on Spoken Language Processing (ICSLP 2002), Denver, USA (2002)
12. Wessel, F., et al.: Confidence measures for large vocabulary continuous speech recognition. IEEE Transactions on Speech and Audio Processing 9, 288–298 (2001)
13. Psutka, J.V., et al.: Searching for a robust MFCC-based parameterization for ASR application. In: SIGMAP 2007: Proceedings of the Second International Conference on Signal Processing and Multimedia Applications, pp. 196–199. INSTICC Press, Lisbon (2007)
14. Young, S., et al.: The HTK Book (for HTK Version 3.4), Cambridge (2006)
15. Stolcke, A., et al.: The SRI March 2000 Hub-5 Conversational Speech Transcription System. In: Proc. NIST Speech Transcription Workshop, College Park, MD (May 2000)
16. 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)

# Speaker Recognition from Coded Speech Using Support Vector Machines

Artur Janicki and Tomasz Staroszczyk

Institute of Telecommunication, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
A.Janicki@tele.pw.edu.pl, tomaszstaroszczyk@gmail.com

**Abstract.** We proposed to use support vector machines (SVMs) to recognize speakers from signal transcoded with different speech codecs. Experiments with SVM-based text-independent speaker classification using a linear GMM supervector kernel were presented for six different codecs and uncoded speech. Both matched (the same codec for creating speaker models and for testing) and mismatched conditions were investigated. SVMs proved to provide high accuracy of speaker recognition, however requiring higher number of Gaussian mixtures than in the baseline GMM-UBM system. In mismatched conditions the Speex codec was shown to perform best for creating robust speaker models.

**Keywords:** speaker recognition, speaker classification, speech coding, support vector machines.

## 1 Introduction

A speaker recognition system often is supposed to analyze voices of remote speakers; this is why the speech signal needs to be transmitted. This in turn implies that the recognition system has to analyze a signal which has been transcoded using one of the existing speech codecs. This can happen in a speaker identification or verification system working in the client-server architecture, where the speech signal is transmitted to the system over the Internet. Also a speaker verification system in a bank should work robustly regardless of the fact that the customer is calling using a land line, a mobile or an Internet phone. This shows why there is a need to make speech recognition robust not only against a change of the microphone or against speaker's inter-session variability, but also against various speech codecs used in voice transmission.

### 1.1 Impact of Speech Coding on Speaker Recognition

Several studies have already been conducted on speaker recognition from coded speech. In majority of cases researches used speaker recognition based on Gaussian mixture models, where speaker models were adapted (using e.g. MAP - maximum a posteriori algorithm) from a universal background model (GMM-UBM systems) [1]. Usually two cases are considered:

- matched conditions - when the speaker recognition system trained using speech transcoded with codec X is tested on speech transcoded with codec X;

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 291–298, 2011.

- mismatched conditions - when the system trained using speech transcoded with codec X (or uncoded at all) is tested on speech transcoded with codec Y.

So it was for example in [2] where the authors showed for the NIST 1998 speaker recognition evaluation corpus how much the recognition accuracy is affected by transcoding using GSM 06.10, G.723.1 and G.729 codecs. The authors reported that GSM 06.10 codec had best results both for matched and mismatched conditions, whilst G.723.1 proved to be the worst (EER rose from 4% to 12% for female speakers), so the performance degradation was consistent with decreasing perceptual quality.

GSM speech codecs were examined in [3], however only in matched conditions. The authors showed that both speaker identification and verification performance is degraded by these codecs, blaming the low LPC order in these codecs for that. They reached the speaker classification accuracy of 68.5% for GSM 06.10 and 71.8% for GSM 06.60.

Speaker recognition from speech coded with GSM 06.60, G.729, G.723.1 and MELP codecs was researched in [4], both for matched in mismatched conditions. The authors used GMM-UBM technique, with gender-dependent UBM models. They found that the recognition accuracy decreases when the mismatch between the quality of ”training” and ”testing” codecs increases. It was shown that using handset dependent score normalization (HNORM) improved the results, especially in mismatched conditions.

Speaker identification from speech transcoded with GSM 06.60 codec was described as well in [5]. The researchers were classifying 60 speakers pronouncing 10 digits in Arabic, recorded in the ARADIGIT corpus. They obtained the identification error rate of 21.94%. In [6] the researchers examined Speex codec, using their own created speech corpus. In various experiments they showed, among others, that Speex can serve well also for creating speaker models for testing GSM-encoded speech. Several studies investigated the possibility to recognize speakers directly from codec's parameters (e.g. [2], [7]), however with results still inferior to those achieved by analysis of the synthetic (transcoded) speech.

### 1.2 SVMs and Speaker Recognition

Support vector machines started to be used in speaker recognition in the middle 90s [8], soon after detailed description of SVMs appeared in [9]. Since that time they have been used successfully in many studies. In [10] the authors used SVMs with Fisher kernel and LR (likelihood ratio) kernel with spherical normalization. On the PolyVar speech corpus they achieved up to 33% relative improvement of speaker verification accuracy compared to GMM-UBM systems. In [11] the authors proposed using an SVM machine to classify supervectors containing GMM parameters (more precisely: Gaussian mixture mean values). They used the linear Kullback-Leibler kernel, which for $M$ Gaussian components can be expressed as:

$$K(utt_a, utt_b) = \sum_{i=1}^{M}(\lambda_i \mu_i^a \Sigma_i^{-1} \mu_i^b) = \sum_{i=1}^{M} \left(\sqrt{\lambda_i}\Sigma_i^{-\frac{1}{2}}\mu_i^a\right)^T \left(\sqrt{\lambda_i}\Sigma_i^{-\frac{1}{2}}\mu_i^b\right) \quad (1)$$

where $\lambda$, $\mu$ and $\Sigma$ are the $i$-th Gaussian parameters (weight, mean values and covariance matrix) of the utterance $a$ and $b$. They proposed also another kernel, called GMM L2 inner product. The authors showed DET curves for a NIST SRE 2005 task, where SVMs outperformed classical GMM ATNorm approach. They also mentioned considerably less computational complexity of the SVM approach.

In [12] the author proposed an improved supervector approach, which was faster and used smaller speaker models.

### 1.3 Aims of This Study

Following promising results of SVM-based speaker recognition in several studies, we decided to apply it for coded speech. In this study we wanted to investigate the following:

- How well SVM-based speaker classification will perform for coded speech?
- What classification accuracy can we achieve with SVMs when recognizing speakers in mismatched conditions?
- Which codec is the best to create speaker models resistant to the mismatch?

Our results will be compared, among others, with the experiments on speaker recognition from speech transcoded with the GSM codecs [3] and the study concerning recognition from Speex-transcoded speech [6].

## 2 Experiment Setup

### 2.1 Speech Data

The TIMIT speech corpus [13] was used as the database of recordings. Although it was originally designed for studies on speech recognition, this corpus was as well used for a number of studies on speaker recognition (e.g. [3], [14]), as it contains recordings of 630 speakers, what is a relatively big number. However, the TIMIT corpus contains single-session recordings only, so the problem of speaker's inter-session variability is not covered in this study.

Each of the speakers utters 10 sentences, each of them lasting 3.2 s on average. Five of these sentences (SX recordings) were used for training the system, whilst the remaining five ones (SA and SI sentences) were used for testing. The SA sentences are the same for every speaker, but they were used in the testing part only, so the text-independency of speaker recognition was preserved. The audio material per a single speaker is relatively short (ca. 32 s, in total for training and for testing, compared e.g. to 120 s in [6]), what makes the TIMIT speaker classification problem even a bigger challenge.

### 2.2 Tested Codecs

In our experiments we decided to recognize speech transcoded with the codecs mostly used in the Internet telephony:

- G.711 (PCM) - used in fixed telephony, but also in VoIP. A-law option was used in this study.

- G723.1 - a codec based on MP-MLQ and ACELP, here the option with the bitrate of 6.4 kbps was used.
- GSM 06.10 (known also as GSM Full-Rate) - designed in the early 90's for the GSM telephony, but used in VoIP as well. Bitrate: 13 kbps.
- GSM 06.60 (known also as GSM-Enhanced Full Rate) - is an enhanced version of GSM 06.10, offering 12.2 kbps bitrate.
- G.729 - operates at a bit rate of 8 kbps, and is based on CS-ACELP. Used in VoIP especially when limited bandwidth is available.
- Speex - a CELP-based lossy codec used in VoIP, offering 10 compression levels at the bitrates of 2.15 - 24.6 kbps (level 8 was used in this study, as it showed the best performance in mismatched conditions in [6]).

### 2.3 Classification

We decided to use a hybrid SVM-GMM approach: we used the SVM algorithm (discriminative part) to classify supervectors, which were made of GMM parameters (generative part). This way, following successful examples, such as [11] and [12], we hoped to benefit of the advantages of both the discriminative and generative approach. A UBM model was trained using 200 speakers and the remaining 430 ones were used for classification experiments, similarly as in [3]. The speech files were parameterized using 19 MFCC parameters (plus the '0' one), with the frame length of 30 ms and 10 ms analysis step. UBM models were created separately for speech transcoded with each codec, using the GMM algorithm with various numbers of mixture components.

The speaker models were created by adapting the UBM model using MAP algorithm with the relevance factor RF = 1. Only mean values of the Gaussian components were adapted, the weight vector and the covariance matrix were not modified. The adapted mean values of each Gaussian mixture were put in a column, thus forming high-dimensional supervectors (SVs). Since we had ca. 16 s of training speech material (SX recordings), we could either create 1 SV per speaker using all of it, or we could split the signal into equal parts and create several training SVs per speaker. Initial experiments showed that using 8 SVs per speaker yielded good classification results, so this number was used in further experiments. Higher number of SVs per speaker sometimes caused that the SVs were not getting enough training data.

We decided to assess the classification accuracy by classifying each of the tested sentences separately. The same procedure of generating SVs was followed for each of the test sentences, so as a result 430 x 5 = 2150 supervectors were created for testing. Classification accuracy was determined by the ratio of correctly classified sentences against the total number of test recordings (i.e. 2150).

The experiments were run in Matlab environment using libsvm toolbox for SVM classification [16] and h2m toolbox for GMM training [17]. The actual classification process was performed using the SVM machine with the classical Kullback-Leibler kernel. When testing classification in matched conditions, the UBM model, training and tested SVs were all created from speech transcoded with the same codec. In experiments with mismatched conditions, the UBM and training sequences were trained using speech transcoded with codec X, and tested on SVs created from speech transcoded with codec Y.

## 3 Results

Before experiments with speaker recognition from coded speech started, first classification was run for speech of original quality ($fs$ = 16 kHz). It turned out that for the number of Gaussian mixtures $M$ = 16 the achieved accuracy was worse than in [3] - it was only slightly over 89% compared to 97.8% in the study using GMM-UBM approach and $M$ = 16, too. When we tried speaker classification from coded speech, it turned out that the same was e.g. for speech transcoded with GSM 06.10 codec: 58.5% compared to 68.5% in [3].

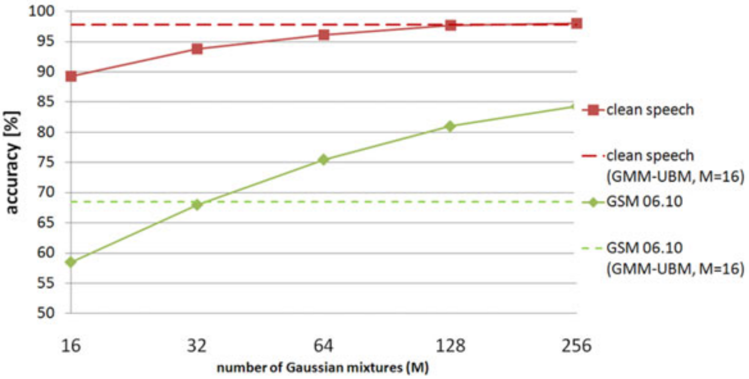


**Fig. 1.** Classification accuracy with SVM-GMM classifier for different numbers of Gaussian components, for clean and coded speech (GSM 06.10 codec). Dashed horizontal lines show recognition results of a GMM-UBM system, reported in [3].

So we decided to increase $M$. After doubling it, the result for GSM 06.10 codec using the SVM-GMM approach was almost the same as for the GMM-UBM algorithm - it reached 68%, see Fig. 1. However due to unsatisfactory result for clean speech we decided to increase further the number of Gaussian components. For $M$ = 256 the accuracy of speaker classification for clean speech outperformed the result achieved in [3] for GMM-UBM classification. What is even more important, the accuracy of classification for coded speech (here for GSM 06.10 codec) increased significantly, from 58.5% to almost 85%, and the distance between performance for uncoded and coded speech decreased from over 30% relative to less than 14%; so we concluded that the gap between the performance for clean and coded speech will be lesser for higher $M$. Thus, in further experiments we decided to use speaker modeling with 256 Gaussian mixtures.

Next experiments were run to check how speaker recognition using SVM-GMM approach is affected by the coded speech. First, the matched conditions were provided, i.e. the system was tested with sentences encoded with the same codec as used during the model training. The performance of speaker classification using the SVM-GMM technique was consistent with speech quality offered by the given codec. The best quality codecs (G.711, Speex - even operating at quality level 8) yielded the best recognition results (88.2% and 86.7%, respectively), while G.723.1 and G.729 proved to be the worse

**Table 1.** Speaker recognition accuracy [%] for systems trained (in rows) and tested (in columns) with different codecs. The diagonal (in bold) shows results for matched conditions.

| training/testing | uncoded | G.711 | G.723 | G.729 | GSM 06.10 | GSM 06.60 | Speex 8 | average | stddev |
|---|---|---|---|---|---|---|---|---|---|
| uncoded | **89.67** | 87.40 | 49.49 | 51.49 | 51.49 | 51.44 | 83.63 | 66.37 | 17.59 |
| G.711 | 86.42 | **88.23** | 46.98 | 47.02 | 50.47 | 64.65 | 80.60 | 66.34 | 16.07 |
| G.723.1 | 71.63 | 68.88 | **73.81** | 61.63 | 60.33 | 71.30 | 75.58 | 69.02 | 4.64 |
| G.729 | 65.16 | 62.37 | 57.95 | **77.12** | 37.72 | 77.91 | 62.74 | 63.00 | 8.91 |
| GSM 06.10 | 69.63 | 70.98 | 55.26 | 44.98 | **83.12** | 57.72 | 72.23 | 64.84 | 10.45 |
| GSM 06.60 | 72.00 | 65.54 | 63.07 | 63.21 | 41.86 | **84.28** | 63.07 | 64.72 | 7.90 |
| Speex 8 | 86.60 | 84.23 | 62.88 | 53.07 | 62.79 | 67.67 | **86.65** | 71.99 | 11.87 |

in this trial, with the recognition accuracy of 73.8% and 77.1%, respectively (see the diagonal in Table 1). Similar relation between speaker recognition performance and codec quality was reported also in other studies, using other classification techniques (e.g. in [2], [6]). The decrease of performance when changing from speech sampled with 16 kHz (98% accuracy) to narrow-band speech sampled with 8 kHz (89.7% accuracy) is noteworthy, too.

Finally, the speaker classification performance was checked under mismatched conditions. As expected, the accuracy decreased here, however not uniformly. The decrease was more significant when there was a remarkable mismatch between ”training” and ”testing” codecs. Testing the system trained with speech transcoded using a high quality G.711 codec with uncoded speech or speech transcoded using another high quality codec (e.g. Speex) results in only minor decrease of accuracy (see Table 1). On the contrary, testing it with the lower-quality G.723.1 or G.729 codecs makes the recognition accuracy almost twice worse. Accuracy in G.723.1/GSM06.60 mismatch is hardly affected (the decrease from 73.8% to 71.3%), as both codecs are CELP-based and offer similar, slightly lower speech quality. This phenomenon had been previously reported in [4] for GMM-UBM-based recognition.

Fig. 2 can help to choose a codec which could be most suitable to create speaker models which would be resistant to mismatched conditions. The line corresponding to Speex is close to G.711 and ”uncoded” lines for high-quality codecs, but also is quite high for the remaining ones, so this codec seem to be the best candidate. This is confirmed also by the highest average recognition rate shown in Table 1 and is consistent with the results reported in [6] for GMM-UBM classification. G.723.1 shows good results here, too - for some codecs (GSM 06.60, G.729) the results for speaker models created G.723.1 outperform the ones created with Speex. The results of G.723.1 are also the most stable for all the tested codecs, so the standard deviation of accuracy is here the lowest (less than 5%). In general G.711 behaves similarly to uncoded speech, due to the simple nature of G.711. The result for G.723.1/Speex8 mismatch is somewhat surprising - it turned out that this result is higher than in the matched conditions (75.6% vs. 73.8%). Also G.729 model tested with GSM 06.60 speech performs surprisingly well. These cases require further investigation.

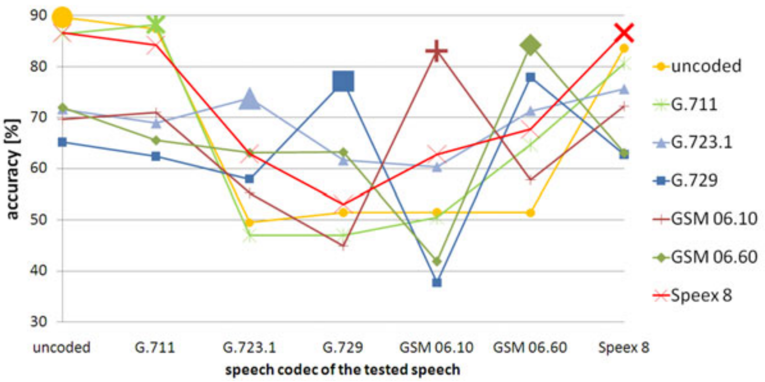


**Fig. 2.** Recognition accuracy for speaker models created with different codecs against the codec of the tested speech. The enlarged markers denote matching conditions.

## 4 Conclusions and Future Works

Our experiments showed that the SVM-based speaker classification from coded speech yields high accuracy results, so it can be used for that purpose. However, the used SVM-GMM technique required higher number of Gaussian mixtures than it was in the baseline GMM-UBM system [3]. This can be explained by the fact, that in a classical GMM-UBM approach the parameters of a tested speech signal are verified against the speaker model, while in the used SVM-GMM technique the decision is based on the model of the tested speech, as the SVM is in fact discriminating the speaker models. This is why used speaker modelling should be more precise to achieve similar results.

Similarly as in other studies the speaker classification accuracy was consistent with codec quality, so Speex and G.711 showed best results. In mismatched conditions the degradation of recognition accuracy is less significant if the ”training” and ”tested” codecs offered similar speech quality.

Future works can explore the possibility of using channel compensation techniques for SVMs, such as NAP, to decrease the impact of coder’s mismatch.

## References

1. Reynolds, D.: Gaussian Mixture Models. Encyclopedia of Biometric Recognition (2008)
2. Quatieri, T., Singer, E., Dunn, R., Reynolds, D., Campbell, J.: Speaker and Language Recognition Using Speech Codec Parameters. In: Proc. Eurospeech 1999, Budapest, vol. 2, pp. 787–790 (1999)
3. Besacier, L., Grassi, S., Dufaux, A., Ansorge, M., Pellandini, F.: GSM Speech Coding and Speaker Recognition. In: Proc. ICASSP, pp. 1085–1088 (2000)
4. Dunn, R., Quatieri, T., Reynolds, D., Campbell, J.: Speaker Recognition from Coded Speech in Matched and Mismatched Conditions. In: Proc. ODYSSEY-2001, Crete, Greece, pp. 115–120 (2001)

5. Krobba, A., Debyeche, M., Amrouche, A.: Evaluation of Speaker Identification System Using GSMEFR Speech Data. In: Proc. 2010 International Conference on Design & Technology of Integrated Systems in Nanoscale Era, Hammamet, pp.1 - 5 (2010)
6. Stauffer, A., Lawson, A.: Speaker Recognition on Lossy Compressed Speech using the Speex Codec. In: Proc. Interspeech 2009, Brighton, pp.2363-2366 (2009)
7. Moreno-Daniel, A., Juang, B.-H., Nolazco-Flores, J.: Speaker Verification Using Coded Speech. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 366–373. Springer, Heidelberg (2004)
8. Schmidt, M., Gish, H.: Speaker identification via support vector classifiers. In: Proc. IEEE International Conference of the Acoustics, Speech, and Signal Processing ICASSP 1996, Atlanta, USA (1996)
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
10. Wan, V., Renals, S.: SVMSVM: Support Vector Machine Speaker Verification Methodology. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), Hong Kong, vol. 2, pp. 221–224 (2003)
11. Campbell, W., Sturim, D., Reynolds, D.: Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters 13, 308–311 (2006)
12. Anguera, X.: MiniVectors: an Improved GMM-SVM Approach for Speaker Verification. In: Proc. Interspeech 2009, Brighton, UK, pp. 2351–2354 (2009)
13. Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., Zue, V.: TIMIT Acoustic-Phonetic Continuous Speech Corpus., Linguistic Data Consortium, Philadelphia (1993)
14. Yu, E., Mak, M.-W., Kung, S.-Y.: Speaker Verification from Coded Telephone Speech Using Stochastic Feature Transformation and Handset Identification. In: Proc. 3rd IEEE Pacific-Rim Conference on Multimedia, Hsinchu, Taiwan, pp. 598–606 (2002)
15. SoX - Sound eXchange, http://sox.sourceforge.net/
16. Chih-Chung, C., Chih-Jen, L.: LIBSVM – A Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/
17. Cappe, O.: h2m Toolkit, http://www.tsi.enst.fr/~cappe/

# Statistical Analysis of Complementary Spectral Features of Emotional Speech in Czech and Slovak

Jiří Přibil[1,2] and Anna Přibilová[3]

[1] Institute of Measurement Science, SAS, Dúbravská cesta 9, SK-841 04 Bratislava, Slovakia
[2] Institute of Photonics and Electronics, Academy of Sciences CR, v.v.i.,
Chaberská 57, CZ-182 51 Prague 8, Czech Republic
Jiri.Pribil@savba.sk
[3] Institute of Electronics and Photonics,
Faculty of Electrical Engineering & Information Technology,
Slovak University of Technology, Ilkovičova 3, SK-812 19 Bratislava, Slovakia
Anna.Pribilova@stuba.sk

**Abstract.** Several spectral features quantify speaker-dependent as well as emotion-dependent characteristics of a speech signal. It means these features provide information which complements the vocal tract characteristics. This paper analyzes and compares complementary spectral features distribution (spectral centroid, spectral flatness measure, Shannon entropy) of male and female acted emotional speech in Czech and Slovak languages.

**Keywords:** spectral features of speech, emotional speech, statistical analysis.

## 1 Introduction

Emotional state of a speaker is accompanied by physiological changes affecting respiration, phonation, and articulation. These acoustic changes are transmitted to the ears of the listener and perceived via the auditory perceptual system [1]. Different types of emotions are manifested not only in prosodic patterns (F0, energy, duration) and several voice quality features (e.g. jitter, shimmer, glottal-to-noise excitation ratio, Hammarberg index) [2] but also by significant changes in spectral domain [3]. Apart from the basic prosodic features, several features related to F0 contour are used for emotion identification: average F0, F0 standard deviation, F0 average variation, minimum F0, maximum F0, and F0 range [4]. As regards the spectral characteristics, they correspond to the shape of the vocal tract which is modified by emotional states. The vocal tract may be characterized by formants representing vocal tract resonances, vocal tract cross-section areas of a concatenated lossless tube model, and coefficients derived from frequency transformations (e.g. mel-frequency cepstral coefficients) [5]. Several spectral features (spectral centroid, spectral flatness measure, Shannon entropy) quantify speaker-dependent as well as emotion-dependent characteristics of a speech signal [6]. It means these features provide information which complements the vocal tract characteristics.

This paper describes analysis and comparison of complementary spectral features (CSF) of male and female acted emotional speech in Czech and Slovak languages.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 299–306, 2011.

Obtained statistical results and values will be used in emotional speech transformation (conversion) method based on cepstral speech description, or/and for creation of the database of values for emotional speech classifier based on statistical evaluation approach. Voice and emotional conversion will be done by modification of neutral spectral parameters according to known parameter ratios between male and female voices as well as ratios between emotional and neutral speech.

## 2 Subject and Method

Complementary spectral features can be determined during cepstral speech analysis (see left part of the block diagram in Fig. 1) using absolute value of the fast Fourier transform |*S*(*k*)| of the speech signal *x*(*n*) and spectral power density *P*(*k*)

$$S(k) = \sum_{n=1}^{N_{FFT}} x(n)\; e^{-j\frac{2\pi}{N_{FFT}}nk}, \quad P(k) = \left|S(k)\right|^2 \Big/ \sum_{k=1}^{N_{FFT}/2} \left|S(k)\right|^2, \tag{1}$$

where $N_{\mathrm{FFT}}$ represents the number of the processed points for FFT calculation.

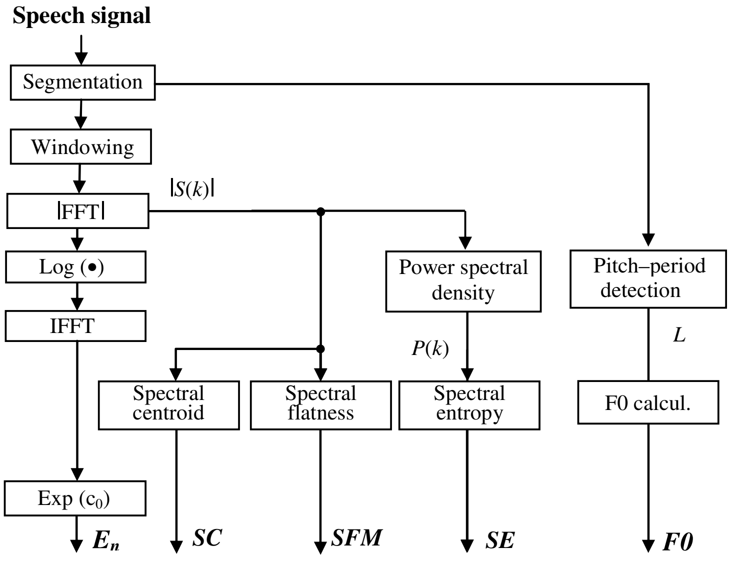


**Fig. 1.** Block diagram of complementary spectral features calculation of the speech signal

The spectral centroid (SC) is a centre of gravity of the power spectrum [6]. It is an average frequency weighted by the values of the normalized energy of each frequency component in the spectrum. It is a measure of spectral shape and "brightness" of the spectrum (higher centroid values correspond to "brighter" voice with more high frequencies). The SC in [Hz] can be calculated as

$$SC = \frac{f_s}{N_{FFT}} \cdot \left( \sum_{k=1}^{N_{FFT}/2} k \left| S(k) \right|^2 \Big/ \sum_{k=1}^{N_{FFT}/2} \left| S(k) \right|^2 \right) . \tag{2}$$

According to psychological research of emotional speech different emotions are accompanied by different spectral noise [1]. In cepstral speech synthesis the spectral flatness measure (SFM) was used to determine voiced/unvoiced energy ratio in voiced speech analysis [7]. This spectral feature can be calculated by the following formula

$$SFM = \left[ \prod_{k=1}^{N_{FFT}/2} \left| S(k) \right|^2 \right]^{\frac{2}{N_{FFT}}} \Big/ \frac{2}{N_{FFT}} \sum_{k=1}^{N_{FFT}/2} \left| S(k) \right|^2 . \tag{3}$$

Spectral entropy is a measure of spectral distribution [8], [9]. It quantifies a degree of randomness of spectral probability density represented by normalized frequency components of the spectrum. Structured speech has lower entropy; non-structured speech has higher entropy. Spectral entropy will be low for spectra having clear formants whereas for unvoiced sounds it will be higher. Shannon spectral entropy (SE) is defined as

$$SE = - \sum_{k=1}^{N_{FFT}/2} P(k) \log_2 P(k) . \tag{4}$$

The values of SC and SFM are calculated only from the voiced speech frames, and in the case of spectral entropy from voiced and unvoiced frames with the signal energy higher than the threshold $\mathrm{En}_{\mathrm{min}}$ – for elimination of speech pauses between words in the sentence and starting and ending parts of the sentence. Calculation of CSF values is supplied with determination of the fundamental frequency F0 and the energy En contour (calculated from the first cepstral coefficient $c_0$ [7]) – see Fig. 1. CSF values obtained by this way are subsequently analyzed separately in dependence on the gender (male / female) and the emotion (joy, sadness, anger, neutral state).

The whole statistical analysis of CSF values consists of six steps:

1. determination of basic statistical parameters,
2. calculation and building of histograms,
3. calculation of extended statistical parameters from histograms,
4. visual comparison of calculated histograms,
5. calculation of the mean values ratios of the CSF for emotional and neutral states,
6. application of the analysis of variances (ANOVA) with multiple comparison of group means, and numerical matching by a hypothesis test.

## 3 Material, Experiments, and Results

The complementary spectral features depend on a speaker as well as the emotions of a speaker [6]. Therefore we collect two speech databases containing neutral and emotional sentences uttered by several speakers (separately from male – 134 sentences, and female voice – 132 sentences, 8+8 speakers altogether) extracted from the Czech and Slovak stories performed by professional actors. Processed speech material consists of sentences with duration from 0.5 to 5.5 seconds, resampled at

16 kHz representing four emotional states (sad, joyful, angry, and a neutral one for comparison). The frame length depends on the mean pitch period of the processed signal – we had chosen 24-ms frames for the male voices, and 20-ms frames for the female voices. The F0 values (pitch contours) were given by autocorrelation analysis method; the energy threshold $En_{min}$ was experimentally set to 0.015 for both voices.

Results of the performed statistical analysis are structured to sets corresponding to the type of the analyzed feature. Every set of three results consists of:

- box-plot graphs of basic statistical parameters – see Fig. 2a, 2b, Fig. 4a, 4b, and Fig. 6a, 6b,
- graphs of filtered histograms – see Fig. 2c, 2d, Fig. 4c, 4d, and Fig. 6c, 6d,
- graphs with visualization of differences between group means calculated using ANOVA statistics – see Fig. 3a, 3b, Fig. 5a, 5b, and Fig. 7a, 7b,
- coupled tables with corresponding resulting null hypothesis/probability values for 5% significance level of the Ansari-Bradley test – see Tables 1, 2, 3.

Results of CSF values extended statistical analysis – skewness and kurtosis[1] parameters determined from histograms for male / female voice in neutral and emotional states are stored in Table 4. The summary results – CSF value ratios between different emotional states and a neutral state for male and female voice – are given in Table 5.

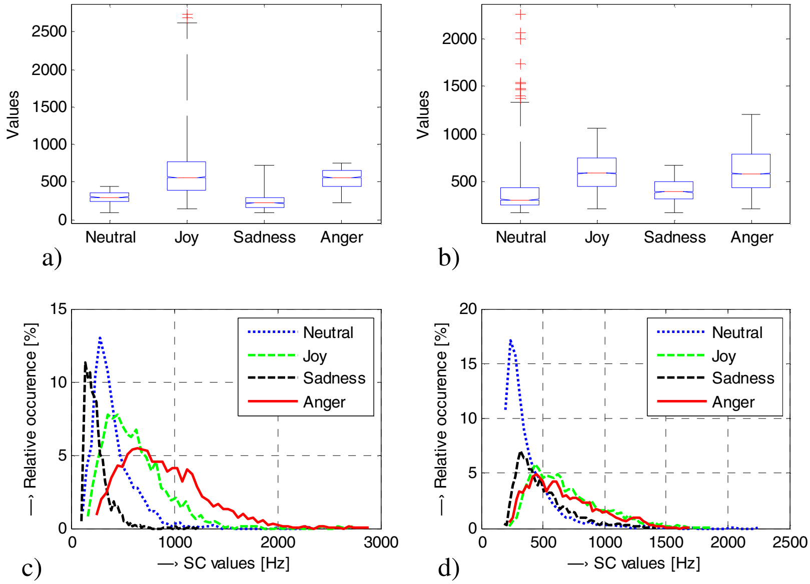


**Fig. 2.** Results analysis of SC values for different speech styles: box plot of basic statistical parameters – male voice (a), and female voice (b); histograms of SC values – male voice (c), and female voice (d), determined from voiced frames only

[1] We use definition of kurtosis which subtracts 3 from the computed value, so that the normal distribution has kurtosis of 0.

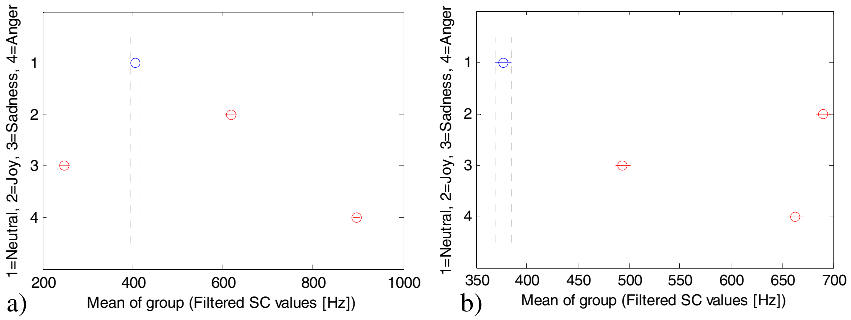


**Fig. 3.** Differences between group means with the help of ANOVA statistics: male voice (a), female voice (b) – SC values

**Table 1.** Results of the Ansari-Bradley hypothesis test of SC values

| **h/p** | **Male voice** | | | **Female voice** | | |
|---|---|---|---|---|---|---|
| | Joy | Sadness | Anger | Joy | Sadness | Anger |
| Neutral | 1/1.98 $10^{-20}$ | 1/2.04 $10^{-16}$ | 1/5.14 $10^{-38}$ | 1/1.33 $10^{-48}$ | 1/1.77 $10^{-17}$ | 1/1.32 $10^{-22}$ |
| Joy | **0/1** | 1/1.65 $10^{-31}$ | 1/8.04 $10^{-30}$ | **0/1** | 1/4.45 $10^{-37}$ | **1/0.002** |
| Sadness | – | **0/1** | 1/7.55 $10^{-43}$ | – | **0/1** | 1/7.18 $10^{-42}$ |

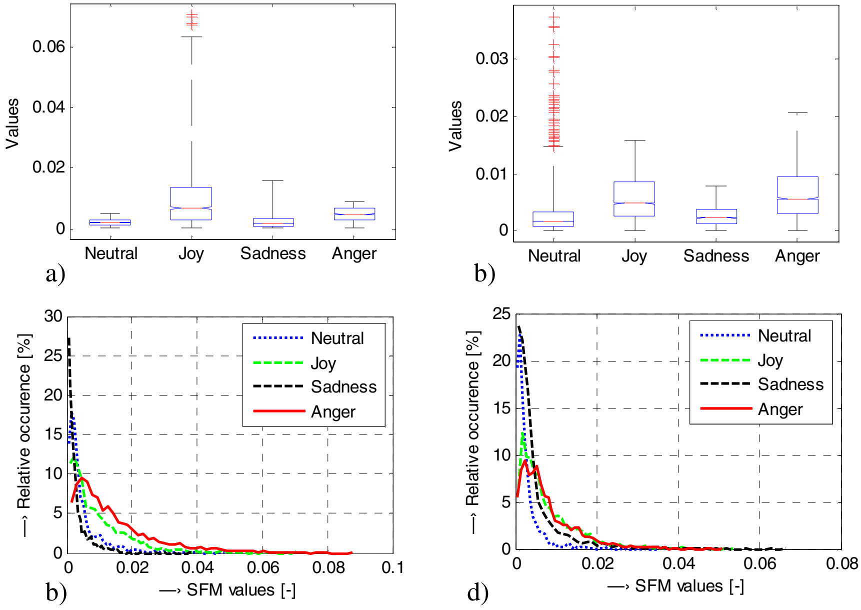


**Fig. 4.** Results analysis of SFM values for different speech styles: box plot of basic statistical parameters – male voice (a), and female voice (b); histograms of SFM values – male voice (c), and female voice (d), determined from voiced frames only

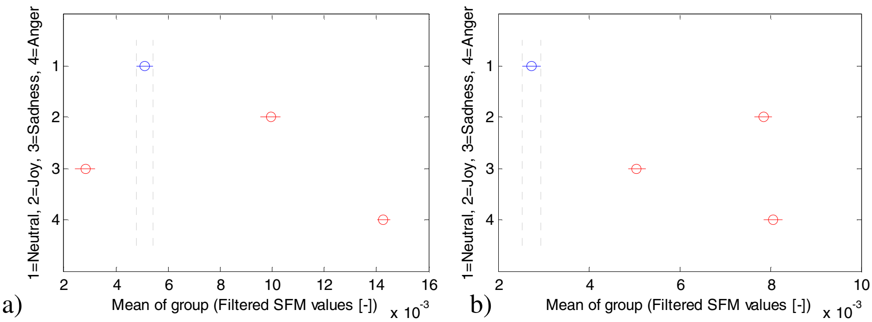


**Fig. 5.** Differences between group means with the help of ANOVA statistics: male voice (a), female voice (b) – SFM values

**Table 2.** Results of the Ansari-Bradley hypothesis test of SFM values

| **h/p** | **Male voice** | | | **Female voice** | | |
|---|---|---|---|---|---|---|
| | Joy | Sadness | Anger | Joy | Sadness | Anger |
| Neutral | 1/1.52 10$^{-12}$ | 1/1.18 10$^{-30}$ | 1/1.08 10$^{-55}$ | 1/1.56 10$^{-12}$ | 1/3.28 10$^{-45}$ | 1/1.24 10$^{-46}$ |
| Joy | **0/1** | 1/1.71 10$^{-28}$ | 1/4.75 10$^{-24}$ | **0/1** | 1/1.32 10$^{-15}$ | **1/0.042** |
| Sadness | – | **0/1** | 1/2.48 10$^{-67}$ | – | **0/1** | 1/5.15 10$^{-24}$ |

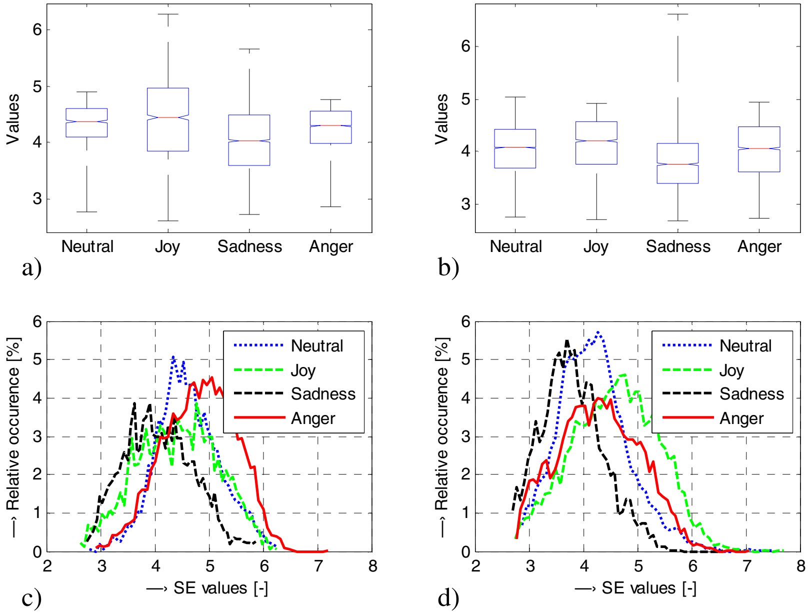


**Fig. 6.** Results analysis of SE values for different speech styles: box plot of basic statistical parameters – male voice (a), and female voice (b); histograms of SE values – male voice (c), and female voice (d), determined from voiced and limited unvoiced frames

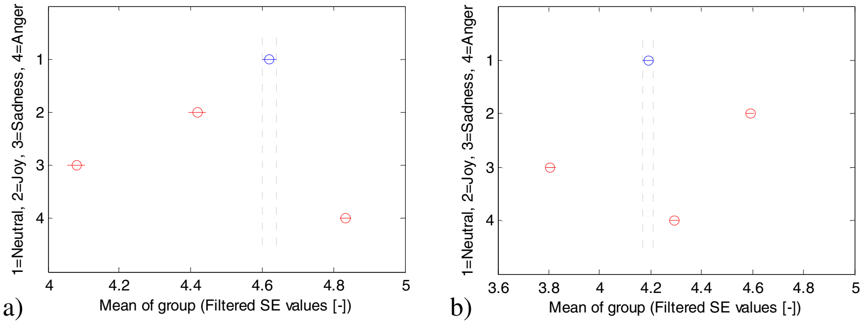


**Fig. 7.** Differences between group means with the help of ANOVA statistics: male voice (a), female voice (b) – SE values

**Table 3.** Results of the Ansari-Bradley hypothesis test of SE values

| **h/p** | **Male voice** | | | **Female voice** | | |
|---|---|---|---|---|---|---|
| | Joy | Sadness | Anger | Joy | Sadness | Anger |
| Neutral | 1/7.68 $10^{-17}$ | 1/3.74 $10^{-35}$ | 1/2.02 $10^{-25}$ | 1/2.44 $10^{-15}$ | 1/2.05 $10^{-7}$ | **0/0.0014** |
| Joy | **0/1** | 1/2.17 $10^{-48}$ | 1/3.28 $10^{-36}$ | **0/1** | 1/5.79 $10^{-44}$ | 1/7.81 $10^{-28}$ |
| Sadness | – | **0/1** | 1/4.52 $10^{-44}$ | – | **0/1** | 1/1.32 $10^{-27}$ |

**Table 4.** Results of CSF values extended statistical analysis: skewness / kurtosis parameters determined from histograms

| **Emotion** | **Male voice** | | | **Female voice** | | |
|---|---|---|---|---|---|---|
| | **SC** | **SFM** | **SE** | **SC** | **SFM** | **SE** |
| Neutral | 2.52 / 7.56 | 2.56 / 8.42 | 0.03 / -0.56 | 2.91 / 7.32 | 3.85 / 12.58 | -0.27 / -0.49 |
| Joy | 1.45 / 2.41 | 2.10 / 5.47 | -0.41 / -0.38 | 0.99 / 0.72 | 2.02 / 5.63 | -0.41 / -0.16 |
| Sadness | 1.52 / 3.29 | 3.28 / 9.30 | -0.24 / 0.08 | 1.39 / 2.04 | 3.32 / 14.79 | -0.31 / -0.04 |
| Anger | 0.92 / 0.83 | 1.84 / 4.71 | -0.76 / 0.41 | 0.81 / 0.12 | 1.75 / 4.05 | -0.61 / 0.20 |

**Table 5.** Summary results of CSF analysis: mean value ratios between different emotional states and a neutral state

| **Parameter** | **joy:** neutral | | **sadness:** neutral | | **anger:** neutral | |
|---|---|---|---|---|---|---|
| **mean ratio** | male | female | male | female | male | female |
| SC | 1.595 | 1.758 | 1.218 | 1.309 | 2.178 | 1.832 |
| SFM | 2.312 | 2.864 | 1.551 | 1.852 | 2.653 | 2.941 |
| SE | 1.064 | 1.094 | 1.056 | 1.048 | 1.107 | 1.097 |

## 4 Conclusion

We performed statistical analysis of CSF values for four emotional states: joy, sadness, anger, and a neutral state. Extended statistical parameters (skewness, kurtosis) were subsequently calculated from these histograms and next they were evaluated by the ANOVA approach and numerical matching by hypothesis tests.

The SC and SFM parameters differ much for voiced and unvoiced speech; therefore only voiced frames of speech signal were analyzed. It is not valid in the case of spectral entropy; hence all speech frames can be analyzed here. Statistical comparison confirms correlation of results for male and female voices, and significant differences between emotional and neutral style data groups. Some statistical "similarity" was observed for female voice between groups "Joy" and "Anger" in the case of SC and SFM parameters (see Fig. 3b and Fig. 5b and Tables 1, 2), and "Neutral" and "Anger" in the case of SE parameter – see Fig. 7b and Table 3. Despite this, obtained results can be used together with values of basic spectral properties and prosodic parameters for creation of the database of values for the emotional speech classifier that is currently being developed.

**Acknowledgments.** The work has been supported by the by the Grant Agency of the Slovak Academy of Sciences (VEGA 2/0090/11), Grant Agency of the Czech Republic (GA102/09/0989), and the Ministry of Education of the Slovak Republic (VEGA 1/0987/12).

## References

1. Scherer, K.R.: Vocal Communication of Emotion: A Review of Research Paradigms. Speech Communication 40, 227–256 (2003)
2. Iriondo, I., Planet, S., Socoró, J.-C., Martínez, E., Alías, F., Monzo, C.: Automatic Refinement of an Expressive Speech Corpus Assembling Subjective Perception and Automatic Classification. Speech Communication 51, 744–758 (2009)
3. Luengo, I., Navas, E., Hernáez, I.: Feature Analysis and Evaluation for Automatic Emotion Identification in Speech. IEEE Transactions on Multimedia 12, 490–501 (2010)
4. Barra, R., Montero, J.M., Marcías-Guarasa, J., D'Haro, L.F., San-Segundo, R., Córdoba, R.: Prosodic and Segmental Rubrics in Emotion Identification. In: Proceedings of 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, pp. I-1085–I-1088 (2006)
5. Ververidis, D., Kotropoulos, C.: Emotional Speech Recognition: Resources, Features, and Methods. Speech Communication 48, 1162–1181 (2006)
6. Hosseinzadeh, D., Krishnan, S.: On the Use of Complementary Spectral Features for Speaker Recognition. EURASIP Journal on Advances in Signal Processing, Article ID 258184,10 (2008)
7. Vích, R.: Cepstral Speech Model, Padé Approximation, Excitation, and Gain Matching in Cepstral Speech Synthesis. In: Proceedings of the 15th Biennial EURASIP Conference Biosignal 2000, Brno, Czech Republic, pp. 77–82 (2000)
8. Li, X., Liu, H., Zheng, Y., Xu, B.: Robust Speech Endpoint Detection Based on Improved Adaptive Band-Partitioning Spectral Entropy. In: Li, K., Fei, M., Irwin, G.W., Ma, S. (eds.) LSMS 2007. LNCS, vol. 4688, pp. 36–45. Springer, Heidelberg (2007)
9. Lee, W.-S., Roh, Y.-W., Kim, D.-J., Kim, J.-H., Hong, K.-S.: Speech Emotion Recognition Using Spectral Entropy. In: Xiong, C., Liu, H., Huang, Y., Xiong, Y. (eds.) ICIRA 2008. LNCS (LNAI), vol. 5315, pp. 45–54. Springer, Heidelberg (2008)

# Statistical-Based Abbreviation Expansion*

Jan Zelinka, Jan Romportl, and Luděk Müller

Department of Cybernetics
University of West Bohemia
306 14, Plzen, Czech Republic
{zelinka,rompi,muller}@kky.zcu.cz

**Abstract.** The work presented in this paper deals with the text normalization for highly inflectional languages. This paper is focused on abbreviation expansion and likewise on numerals normalization. Our text normalization system does not use any explicit parser or part-of-speech tagger and thus it can be called lightly supervised. The standard rule-based text normalization method is compared with the proposed statistical-based one in the task of expansion of Czech abbreviations.

## 1 Introduction

Our paper deals with text normalization for language models constructions in automatic speech recognition systems or for text-to-speech system construction. We expect from a text normalization system that it can the following tasks: typos correction, transformation of relevant words to lower case, transformation of numerals into their word forms and finally abbreviation expansion. This particular paper is focused on abbreviation expansion but we also tested methods on numerals normalization in our experiments.

In general, abbreviation expansion is a very difficult task because two different words can have (and often do have) a very same abbreviated form and the original words can be estimated only by means of their context. There is another problem in Slavic languages such as Czech (which the paper is focused on) because there are usually many forms of a noun or an adjective due to the flective nature of these languages. Such a problem of word form ambiguity in these languages can be even more complicated than word ambiguity in languages like English. For example, full forms for the abbreviation "obv." are "obvodní", "obvodního", "obvodnímu", "obvodním", "obvodních" (i.e. adjective "district" as in "district council" with all its grammatical cases), ..., "obvyklý", "obvyklá", "obvyklé", "obvyklého", "obvyklou", "obvyklému" (i.e. adjective "common" as in "common price" with all its grammatical cases), ..., "obvazový", "obvazové", "obvazového", "obvazovému", "obvazovém" (i.e. adjective "dressing" as in "wound dressing material" with all its grammatical cases), ..., etc.

The complexity of this task leads us to the opinion that the task cannot be solved by means of a system with several number of handwritten rules. Therefore, machine learning methods were adopted for the abbreviation expansion task. We can say that

* This research was supported by the Grant Agency of the Czech Republic, project No. GAČR 102/08/0707 and the Technology Agency of the Czech Republic, project No. TA01011264 and the Ministry of Education of the Czech Republic, project No. MŠMT LC536.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 307–314, 2011.

standard abbreviation expansion methods are modified methods for automatic translation or part-of-speech tagging. A modified Brill tagger was tested in our experiments as a standard method, and as a novel approach we introduced here a statistical-based expansion method.

A training data set is necessary for every machine learning method. Our data set consists of medical documentation texts [1]. Unfortunately, there is insufficient number of "manually" (i.e. by a human annotator) expanded abbreviations in our corpus, and therefore a simple automatic method which expands abbreviations only in credible cases is necessary. We suppose that there are many attentively written whole (i.e. unabbreviated) words in the text which are abbreviated elsewhere, and the only expert information available for us is a list of abbreviations (even this list can be easily generated automatically but accuracy of such a process is very questionable).

Hereinafter we shall call *context* a couple given for a particular word as its immediately succeeding and its immediately preceding word. During the data set construction, contexts of abbreviations are sought, as well as whole words in these contexts which can be abbreviated into the corresponding abbreviations. These whole words are then considered to be our expanded abbreviations. Only abbreviation with length higher than 3 characters are expanded this way because expanding abbreviations shorter than 4 is much less reliable. The whole abbreviation expansion process is shown in Fig. 1.

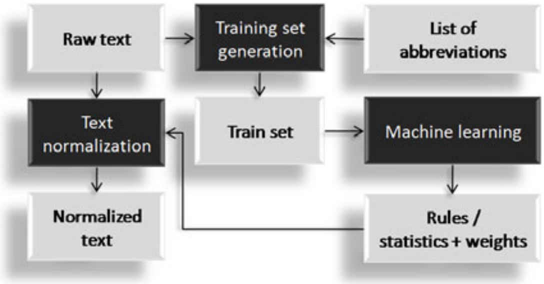


**Fig. 1.** A scheme of text normalization process

## 2 Rule-Based Abbreviation Expansion

In our experiments, we used a special induction expert system for each abbreviation. Rules of the described expert system are of the following form:

$$(LC \equiv LC_{re} \: \& \: RC \equiv RC_{re}) \rightarrow output = O.$$

It means that if a left context $LC$ of an abbreviation is equivalent to $LC_{re}$ and a right context $RC$ of the abbreviation is equivalent to $RC_{re}$, then a word form of the abbreviation is $O$. $LC_{re}$ and $RC_{re}$ are regular expresions, i.e. $a \equiv b$ means that the regular expression $b$ matches the string $a$. Regular expressions used in our algorithm are only of these two forms: 1) $\hat{\ } a_1 a_2 \dots a_n\$$, where $a_i \in \{''a'', \dots, '' z''\}$ and $n > 0$ is not restricted, 2) $\hat{\ }. + a_1 a_2 \dots a_n\$$ where $n = 0, 1, 2, 3, 4, 5$ (i.e. the regular expression $\hat{\ }. + \$$

matches with any word is also in the consideration). We must note that ˆ and $ mean the first position and the final position of words, not of the whole sentence. The set of rules is ordered and the rules are applied sequentially.

The used rule-based abbreviation expansion system is, in principle, a modified Brill tagger, i.e the rule induction process is an iterative process which operates in the following steps:

1. The set or rules is empty.
2. Set all expansions in the data set as the most probable expansions in the data set.
3. Find the rule $r$ with the maximal evaluation $c(r) = c_+(r) - c_-(r)$, where $c_+$ is the number of correct expansions which the rule provides and $c_-$ is the number of incorrect expansion provided by the rule.
4. If $c(r) > 0$ (or the number of iteration exceeded a given threshold), then add the rule into the set of rules, apply the rule on the data set and skip to the step 3.

All the possible rules which match with at least one example in the data set can be effectively generated and evaluated. Hence, this algorithm can be used even in case of a very large data set.

## 3 Statistical-Based Abbreviation Expansion

No variable for a numeral is introduced below because we constructed a special statistical-based expansion system for each abbreviation.

The resulting expansion in the statistical-based expansion system is a word $\hat{w}(LC, RC, \lambda) = \arg\max_w p(w|LC, RC, \lambda)$, where $p(w_i|LC, RC, \lambda)$ is the used discriminative function for a word form $w$ (which can be seen as a posterior probability estimate) and it is given by the formula:

$$p(w|LC, RC, \lambda) = \sum_{i=1}^{n} \lambda_i S_i(w|LC, RC), \qquad (1)$$

where $n \in N$, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ and $S_i$ is the $i$-th used statistics. For the statistics computation in our statistical-based expansion system, $n$ context transformations $CT_1, CT_2, \ldots, CT_n$ and $n$ abbreviation transformations $WT_1$, $WT_2$, …, $WT_n$ were applied. A context transformation $CT_i$ transforms a couple $(LC, RC)$ (where $LC$ is the left context and $RC$ is the right context) into a couple $(LC', RC')$ and an abbreviation transformation $WT_i$ transforms a word $w$ into a word $w'$. In our experiments, we chose the transformations as follows (for $n = 10$):

$$\begin{aligned}
CT_1(LC, RC) &= (LC, \circ), & CT_2(LC, RC) &= (\circ, RC),\\
CT_3(LC, RC) &= (\circ + end_1(LC), \circ), & CT_4(LC, RC) &= (\circ + end_2(LC), \circ),\\
CT_5(LC, RC) &= (\circ + end_3(LC), \circ), & CT_6(LC, RC) &= (\circ, \circ + end_1(RC)),\\
CT_7(LC, RC) &= (\circ, \circ + end_2(RC)), & CT_8(LC, RC) &= (\circ, \circ + end_3(RC)),\\
CT_{9,10}(LC, RC) &= (\circ, \circ), & WT_{1-9}(w) &= w,\ WT_{10}(w) = \circ
\end{aligned}$$

where $\circ$ is a symbol for a word which does not appear in the data set, $+$ means string concatenation and function $end_m(x)$ returns the last $\min\{m, |a|\}$ characters of a word $x$. The statistics $S_i$ for $i = 1, \ldots, n$ are computed in the following manner:

$$S_i(w|LC, RC) = \frac{\#(WT_i(w), CT_i(LC, RC))}{\#(CT_i(LC, RC))}, \tag{2}$$

where $\#(WT_i(w), CT_i(LC, RC))$ is the frequency of the transformed word within the transformed context in the transformed data set and $\#(CT_i(LC, RC))$ is the frequency of the transformed context in the transformed data set (indeed, $S_i(w|LC, RC) = \frac{1}{n_w}$ when $\#(CT_i(LC, RC)) = 0$ where $n_w$ is the number of word forms).

A training set $S$ consists of the quaternion: abbreviation, left context, right context and the correct expansion. The optimal $\lambda$ for the training set $S$ is a vector which maximizes the accuracy function

$$acc(S, \lambda) = \sum_{(abb, LC, RC, w) \in S,\ w = \hat{w}(LC, RC, \lambda)} 1. \tag{3}$$

Let us have an example $(abb, LC, RC, w) \in S$ and two different words $a, b$. A margin in a space $\Lambda = \{\lambda; \lambda \in R^n\}$ where decision $\hat{w}(LC, RC, \lambda) = a$ might be changed to $\hat{w}(LC, RC, \lambda) = b$ is given by the equation $p(a|LC, RC, \lambda) = p(b|LC, RC, \lambda)$. This equation is equivalent to the equation $\alpha^{\mathrm{T}}(x, a, b)\lambda = 0$, where $\alpha(x, a, b)$ is a vector

$$\alpha(x, a, b) = \begin{pmatrix} S_1(a|LC, RC) - S_1(b|LC, RC) \\ S_2(a|LC, RC) - S_2(b|LC, RC) \\ \vdots \\ S_n(a|LC, RC) - S_n(b|LC, RC) \end{pmatrix}. \tag{4}$$

A set of (hyper) planes

$$\alpha^{\mathrm{T}}(x, a, b)\lambda = 0,\ x \in S,\ a, b \in \{w_1, w_2, \ldots, w_{n_w}\},\ a \neq b \tag{5}$$

divides the space $\Lambda$ into a finite number of convex figures. The accuracy function is constant inside each figure because there is no margin inside any figure on which any classification can be changed [2] [3].

Consequently, to choose one single random point from each figure and to evaluate the points by the criterion $acc(S, \lambda)$ will be enough to find the optimal $\lambda$. Unfortunately, this geometrical problem is trivial only in spaces with dimension 1 or 2. We have decided to design an algorithm which transforms the problem of accuracy maximization in the space with dimension $n$ into a sequence of problems of accuracy maximization in a space with dimension 2. Our proposed algorithm is an iterative process and its initial step is a random vector $\lambda(0)$. In the $k$-th step a random vector $\Delta(k) \in \Lambda \setminus \{0, \lambda_k\}$ is generated. The next $(k+1)$-th estimation is a vector $\lambda(k+1) = \alpha(k)\lambda(k) + \beta(k)\Delta(k)$, where $\alpha(k)$ and $\beta(k)$ are the optimal weights for which the accuracy $acc_{k+1} = acc(S, \alpha(k)\lambda(k) + \beta(k)\Delta(k))$ is maximal. The described process can be obviously only suboptimal, but $acc_k \leq acc_{k+1}$ holds because at worst $acc_k = acc_{k+1}$ for $\alpha = 1$ and $\beta = 0$. Fig. 2 shows the $(k+1)$-th step of our algorithm.

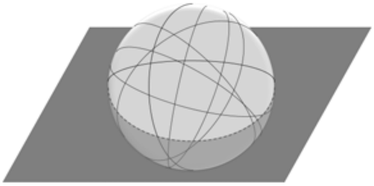


**Fig. 2.** An illustration of $(k + 1)$-th step of our algorithm in a 3D space

Values of $\alpha$ and $\beta$ are restricted on $\alpha^2 + \beta^2 = 1$ because the optimal $\alpha$ and $\beta$ lie also in the unit circle. Consequently, the 2D problem is equivalent with the 1D problem which is very simple in principle. Each separating line can be represented by its intersection with the unit circle $\alpha^2 + \beta^2 = 1$. The intersection consists of two points where only angles are relevant for the separating line representation if the point $p_1$ and $p_2$ are converted into polar coordinates $[1, \varphi_1]$ and $[1, \varphi_2]$ (moreover, obviously $\varphi_1 \equiv -\varphi_2$.) A classification for an angle $\varphi$ and an example $x \in S$ is a classification $\hat{w}(x, \varphi) = \hat{w}(LC, RC, \cos(\varphi)\lambda(k) + \sin(\varphi)\Delta(k))$. For each example $x \in S$ a maximal set of open intervals $Is(x) = \{(\varphi_1, \varphi_2), (\varphi_3, \varphi_4), \ldots, (\varphi_{m-1}, \varphi_m)\}$, where decision $\hat{w}(x, \varphi)$ for $\varphi \in Is(x)$ is correct, can be constructed easily. Our algorithm simply sorts angles $\varphi \in \{\varphi'; (\varphi', \varphi'') \in Is(x) \vee (\varphi'', \varphi') \in Is(x), x \in S\}$ into a sequence $\varphi_1 < \varphi_2 < \varphi_2 < \cdots < \varphi_m$ and it computes accuracies for all points $\frac{\varphi_{i-1}+\varphi_i}{2}$ for $i = 2, 3, \ldots, m$ and the point $\pi$ when there is no boundary in the point. The argument of maximum is the optimal angle $\varphi$. We can compute accuracy directly or we can accelerate it by using the following manner: The accuracy can be computed by means of functions which we call particular accuracies:

$$pacc(x, \varphi) = \begin{cases} 1 & \varphi \in Is(x) \\ 0 & \varphi \notin Is(x) \end{cases} \tag{6}$$

Hence $acc(\varphi) = \sum_{x \in S} pacc(x, \varphi)$. Fig. 3 illustrates this accelerated accuracy maximization process[1].

To avoid the phenomenon of overtraining, the used training set was split into two disjoint sets with the length ratio 9:1. The statistics were computed from the first part of the training set and the weights $\lambda_i$ were estimated from the second part of the training set.

## 4 Experiments and Results

We split our data set (with 208810 examples) into the training and testing sets in ten different ways, so as to obtain more objective assessment of characteristics of both

[1] The maximum of accuracy was our only goal and we did not care to preserve necessary properties of probability distribution, i.e. to guarantee that $\lambda_i \geq 0$ for $i = 1, \ldots, n$ and $\sum_i \lambda_i = 1$ in eq. (1). But if there is no negative weight, the weights can be transformed and the properties of probability distribution will hold and the accuracy will not be changed. Otherwise, there is no possibility for this transformation, and therefore if the properties must be saved, the angles in our algorithm must be restricted on the interval $\left\langle 0, \frac{\pi}{2} \right\rangle$.

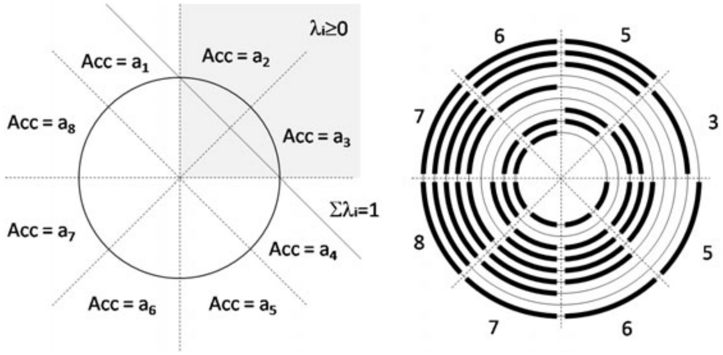


**Fig. 3.** An illustration of accuracy maximization process

methods. The training and testing set ratio size was always 9:1, hence there were 187929 examples in each training set and 20881 in each testing set. Naturally, each training was mutually disjoint with its respective testing set, and furthermore all testing sets were pairwise disjoint too. The observed accuracies for the testing sets are the accuracies given by the case when a context of a particular numeral does not appear in the training set. The results for the rule-based abbreviation expansion system are in Tab. 1 and the results for the statistical-based system are in Tab. 2.

**Table 1.** The results for the rule-based abbreviation expansion system

| #rules | Acc. training set [%] | Acc. testing set [%] |
|---|---|---|
| average | min – average – max | min – average – max |
| 65.0 | 50.6 – 52.7 – 53.9 | 38.4 – 47.8 – 63.0 |
| 127.2 | 60.1 – 62.0 – 64.3 | 40.6 – 53.2 – 66.8 |
| 250.7 | 67.2 – 69.0 – 70.5 | 44.2 – 56.6 – 70.0 |
| 431.0 | 72.7 – 74.0 – 75.2 | 45.9 – 59.3 – 71.7 |
| 649.8 | 76.6 – 77.7 – 78.8 | 58.2 – 65.9 – 73.2 |
| 899.7 | 79.5 – 80.4 – 81.1 | 59.9 – 67.6 – 74.0 |
| 1180.6 | 81.5 – 82.6 – 83.3 | 58.3 – 68.6 – 76.1 |

In our experiments we tested the described method also on numerals normalization problems, for which the data set acquisition is described in [4]. There were 239931 examples in each training set and 26659 in each testing set. The results for the rule-based numerals normalization system are in Tab. 3 and the results for the statistical-based numerals normalization system are in Tab. 4.

Inclination to overtraining is apparent in all cases. The average of the accuracies for the statistical-based method is significantly higher. Also the accuracies are in a smaller interval in the case of the statistical-based method. But above all, the statistical-based method does not fail on some data sets, whereas the rule-based method does (see minima in Tab. 1 and 3).

**Table 2.** The results for the statisical-based abbreviation expansion system

| | Acc. training set [%] | Acc. testing set [%] |
|---|---|---|
| #steps | min – average – max | min – average – max |
| 0 | 50.6 – 52.7 – 53.9 | 38.4 – 47.8 – 63.0 |
| 1 | 81.3 – 88.0 – 91.5 | 70.9 – 76.1 – 79.3 |
| 2 | 81.5 – 87.4 – 91.5 | 71.6 – 76.0 – 80.0 |
| 3 | 81.4 – 87.3 – 91.3 | 72.0 – 76.4 – 80.5 |
| 4 | 81.3 – 87.3 – 91.5 | 71.9 – 76.3 – 80.3 |
| 5 | 81.3 – 87.4 – 91.3 | 72.0 – 76.1 – 80.4 |
| 6 | 81.2 – 87.5 – 91.1 | 72.3 – 76.2 – 80.4 |

**Table 3.** The results for the rule-based numerals normalization system

| #rules | Acc. training set [%] | Acc. testing set [%] |
|---|---|---|
| average | min – average – max | min – average – max |
| 40.0 | 57.0 – 59.5 – 60.6 | 49.5 – 58.4 – 76.0 |
| 80.0 | 64.4 – 66.4 – 67.9 | 53.8 – 63.2 – 78.3 |
| 149.2 | 68.2 – 69.7 – 71.1 | 55.8 – 64.5 – 79.3 |
| 232.3 | 70.9 – 72.3 – 73.6 | 58.5 – 67.2 – 80.1 |
| 324.6 | 73.0 – 74.0 – 74.9 | 56.5 – 68.3 – 81.0 |
| 416.0 | 74.9 – 75.7 – 76.5 | 48.9 – 66.4 – 74.5 |

**Table 4.** The results for the statistical-based numerals normalization system

| | Acc. training set [%] | Acc. testing set [%] |
|---|---|---|
| #steps | min – average – max | min – average – max |
| 0 | 56.1 – 59.2 – 60.6 | 49.5 – 57.1 – 74.5 |
| 1 | 86.0 – 87.2 – 88.7 | 73.8 – 81.0 – 90.6 |
| 2 | 86.2 – 87.5 – 88.9 | 73.5 – 81.4 – 90.9 |
| 3 | 86.2 – 87.7 – 89.0 | 74.5 – 81.7 – 90.9 |
| 4 | 86.3 – 87.8 – 89.0 | 74.5 – 81.8 – 90.9 |
| 5 | 86.4 – 87.9 – 89.1 | 74.3 – 81.7 – 91.0 |
| 6 | 86.4 – 88.0 – 89.2 | 74.3 – 81.8 – 91.0 |

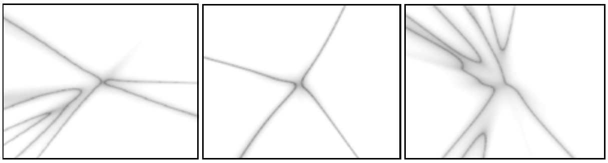


**Fig. 4.** Numerically computed classification changing margins for a random ANN

## 5 Conclusion and Future Work

The described rule-based method is the standard method for text normalization. Another approach can be found in [5]. The main contribution of our work is the parameter estimation method described in Section 3. The results of the presented experimental evaluation clearly prove that this method is beneficial.

In the future, we will modify the presented weight estimation method for artificial neural network (ANN) training. However, separating curves in a plane of tuples $(\alpha, \beta)$ which are given by an equation $y_i(x, \alpha\theta + \beta\Delta) = y_j(x, \alpha\theta + \beta\Delta)$, where $y_i$ or $y_j$ is an $i$-th or $j$-th output of an ANN, $i \neq j$, $x$ is an ANN input and $\theta$ and $\Delta$ are parameters of the ANN, enclose relatively complicated areas, as it is shown in Fig. 4. Yet it is possible to seek through one dimensional space of real numbers $\gamma$, i.e. find all intervals between roots of an equation $y_i(x, \theta + \gamma\Delta) = y_j(x, \theta + \gamma\Delta)$ It is easy to prove that there is a finite number of roots of the equation and moreover, all roots can be found with a given precision.

## References

1. Hippman, R., Dostálová, T., Zvárová, J., Nagy, M., Seydlová, M., Hanzlíček, P., Kříž, P., Šmídl, L., Trmal, J.: Voice-supported electronic health record for temporomandibular joint disorders. Methods of Information in Medicine 49, 168–172 (2010)
2. Caruana, R., Niculescu-Mizil, A.: Data mining in metric space: An empirical analysis of supervised learning performance criteria, pp. 69–78. ACM Press, New York (2004)
3. Shen, Y.: Loss Functions for Binary Classification and Class Probability Estimation. PhD thesis (2005)
4. Sproat, R.: Lightly supervised learning of text normalization: Russian number names. In: IEEE Workshop on Spoken Language Technology, Berkeley, U.S.A (2010)
5. Schlippe, T., Zhu, C., Gebhardt, J., Schultz, T.: Text normalization based on statistical machine translation and internet user support. In: INTERSPEECH, pp. 1816–1819 (2010)

# The Role of Neural Network Size in TRAP/HATS Feature Extraction

František Grézl*

Brno University of Technology, Speech@FIT, Czech Republic
grezl@fit.vutbr.cz

**Abstract.** We study the role of sizes of neural networks (NNs) in TRAP (TempoRAl Patterns) and HATS (Hidden Activation TRAPS architecture) probabilistic features extraction. The question of sufficient size of band NNs is linked with the question whether the Merger is able to compensate for lower accuracy of band NNs. For both architectures, the performance increases with increasing size of Merger NN. For TRAP architecture, it was observed, that increasing band NN size over some value has not further positive effect on final performance. The situation is different when HATS architecture is employed – increasing size of band NNs has mostly negative effect on final performance. This is caused by merger not being able to efficiently exploit the information hidden in its input with increased size. The solution is proposed in form of bottle-neck NN which allows for arbitrary size output.

## 1 Introduction

The neural network (NN) based features are gaining more and more importance in today's ASR systems. Their era started more than a decade ago by introducing the TANDEM approach [1], where outputs of one classifier were treated as features for the second classifier. The first classifier is a Neural Network (NN) (or a structure of several NNs) trained to classify phonetically motivated classes and its outputs are estimates of class probabilities. The second classifier is a standard GMM-HMM system. As probabilities do not have the desired Gaussian distribution, they were usually processed by logarithm and decorrelated by Principal Component Analysis. The resulting features are called probabilistic features.

The TRAP (TempoRAl Pattern) feature extraction was one of the first employed in TANDEM scheme [2,3]. TRAP features are derived from long temporal context (up to 1s) of primary features, mostly outputs of Mel-filter bank critical band energies (CRBE). The temporal evolution of energy in one critical band forms *TRAP vector*. This *TRAP vector* is converged into phoneme probability estimates by its own NN (band NN). This is done for all coefficients/bands. Outputs from all band NNs are concatenated into one vector and, after logarithm nonlinearity, form input to Merger NN.

* This work was partly supported by Technology Agency of the Czech Republic grant No. TA01011328, Czech Ministry of Education project No. MSM0021630528, Grant Agency of Czech Republic projects Nos. GP102/09/P635 and 102/08/0707, and by BUT FIT grant No. FIT-11-S-2.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 315–322, 2011.

Merger NN combines all band-conditioned estimates into one final set of probability estimates.

The performance of probabilistic features in the ASR system is closely tied (although the direct relation does not exists) to the classification accuracy reached by NN during the training. Thus the improvements of probabilistic features were focused on reaching higher classification accuracy of the NN. In the context of TRAP, several proposals addressing different stages of the processing were made. Different ways of *TRAP vector* extraction were for example addressed in [4,5]. Processing of several *TRAP vectors* by one band-NN was evaluated in [6,7]. Different structures of band NNs and Merger were studied in [8]. Finally, one can always play with increasing the NNs size [9].

As there is always the question *"What happens, if you make the NN bigger?"* we would like to address the last point in our analysis. There are two kinds of NNs in TRAP processing - band NNs and Merger and so the analysis can be split into two parts:

- Changing the size of band NN and keeping Merger size constant can tell what classification accuracy can be reached by band NN and how it influences the final accuracy of the Merger. The minimum sufficient size of band NNs can be found in this way.
- Altering Mergers size while keeping the band NNs constant can show how the classification accuracy change having the same input, e.g. what is the maximum accuracy one can reach with given band-NNs accuracy.

Finally, it would be possible to tell to what extent is the merger able to compensate the lower classification accuracy of band NNs and to find optimal sizes of NNs in the architecture.

The effect of altering NNs sizes is not observed only on classification accuracies but also on Word Error Rate (WER) of Large Vocabulary Continuous Speech Recognition (LVCSR) systems on meeting speech recognition as defined by NIST RT'07 speech-to-text evaluations.

## 2 Probabilistic Features

Ideally, we would like such features, that have maximum mutual information between the feature vector $\mathbf{x}$ and the class $Q_i$ they belong to. It has been shown, that maximizing the *aposteriori* probability of class maximizes also the mutual information $I(\mathbf{x}, Q_i)$, under the condition that all classes $Q_i, i = 1 \ldots K$ are equally likely [10].

An ideal feature extraction should be able to reduce the error to its theoretical limit given by Bayes' error [11]. For $K$ class problem, the Bayes classifier compares aposteriori probabilities of vector $\mathbf{x}$ : $p(\mathbf{x}|Q_i)$ for all classes and classifies $\mathbf{x}$ to the class with maximum aposteriori probability. Since aposteriori probabilities are not linearly independent, as

$$\sum_{i=1}^{K} p(\mathbf{x}|Q_i) = 1, \tag{1}$$

only $K - 1$ probabilistic features would be the ideal set of features which would give the Bayes' error.

To estimate class aposteriori probabilities, the discriminative connectionist model – artificial neural network (NN) – is used. This model learns the transform of the input vector $\mathbf{x}$ to aposteriori probability directly from the data.

The discriminative training of the model focuses on the boundary between the classes where the differences are magnified, whereas the details in the "middle" of the class are rather minimized. This transformation makes the resulting probabilistic features more separable. This issue was discussed in [1].

## 3 System Description

The recognition task is meeting speech recognition as defined by the NIST RT'07 STT evaluations. The independent head set microphone (IHM) condition with reference segmentation was used in our experiments.

The **Critical Band Energies** (CRBE) are computed from 25ms of speech every 10ms. The speech signal is sampled at 16 KHz and there are 23 filters in the filter-bank analysis. CRBE are subject to mean and variance normalization on speaker basis.

**Post-processing** of Mergers output consists of logarithm and Heteroscedastic Linear Discriminant Analysis (HLDA) decorrelation and dimensionality reduction to 30 dimensions. The HLDA treats every state of corresponding HMM model as class.

The **Recognition system** is based on AMI-LVCSR used in NIST RT'07 evaluation [12] which is quite complex system running in many passes. For these experiments, the process stopped after the first decoding pass and estimation of VTLN warping factor. The system was simplified by omitting the constrained MLLR adaptation and lattice generation followed by four-gram Language Model (LM) expansion. Full decoding using bi-gram LM was done instead. The LM scale factor and the word insertion penalty estimated on RT'05 were used here.

The **training set** consists of the complete NIST, ISL, AMI and ICSI meeting data – about 180 hours. The NN were trained on subset of 173 hours. The transcription for NN training were obtained by forced alignment of training data using enhanced PLP features [12].

The **features** used in recognition system are the post-processed outputs from Merger only. Although delta parameters or concatenation with cepstral features improve the performance, for the purpose of our analysis, it is better to use only outputs from the system under evaluation.

### 3.1 TRAP/HATS Neural Network Architectures

The concept of **TRAP architecture** was given in Sec. 1, a more detailed description follows. The length of TRAP vector is 51 frames which covers 0.5 second of speech signal. There are 23 band-NNs which are trained towards 45 phoneme classes including silence. All used NNs have three layers. The scheme is shown in Fig. 1.

The **Hidden Activation TRAPS architecture** (HATS) further improved the performance of resulting probabilistic features [13]. As the name suggests, the outputs of band NN hidden neurons (after sigmoid nonlinearity) are taken to create inputs for Merger. The logarithm between the first stage outputs and second stage inputs is omitted.

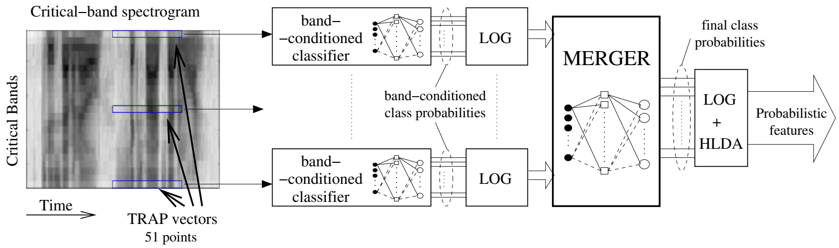


**Fig. 1.** Scheme of basic TRAP architecture

**Table 1.** Frame cross-validation accuracy of $6^{th}$ band NN [%]

| Total band NNs weights | 100 K | 200 K | 500 K | 1 M | 2 M |
|---|---|---|---|---|---|
| neurons in hidden layer | 45 | 90 | 226 | 452 | 904 |
| cross-validation acc | 27.7 | 29.0 | 30.1 | 30.7 | 32.3 |

The numbers of weights assigned to band NNs (sum of weights in all band NNs) were 100K 200K 500K 1M and 2M. The numbers of weights in merger were 1M 1.5M 2M and 3M.

## 4 Experimental Results and Discussion

First, the frame accuracy on cross-validation data[1] of $6^{th}$ band NN is shown together with number of NN hidden units in Tab. 1. The classification accuracy increases with increasing number of weights in NN.

Next, Mergers with different numbers of weights are trained on each set of band NNs. The respective cross-validation accuracies are given in left part of Tab. 2. Then, the LVCSR system is trained on probabilistic features from each Merger and WERs are obtained. See right part of Tab. 2.

The following observations are made from these results:

- the system performance increases with increasing size of the Merger
- increasing band NNs size over 200 K weights does not further increase the performance
- the best system is not the biggest one

These observations suggest that higher classification accuracies of band NNs either cannot be utilized by the Merger, or are not necessary because the Merger is able to obtain the information by combining all band NNs outputs. In both cases, it would be interesting to find out where the band NNs improvements come from. We focused on this issue in the following section.

The HATS architecture was also evaluated. Note that the number of inputs to HATS Merger is changing with changing size of band NNs and thus the number of Mergers

[1] 10% of training data on which the NN is not trained which serves for measuring of improvements and early stopping of NN training.

**Table 2.** The Merger Cross-Validation frame accuracies and LVCSR WERs [%]

Cross-Validation frame accuracies [%]

| band | Merger weights / hidden units | | | |
|---|---|---|---|---|
| NNs | 1 M | 1.5 M | 2 M | 3 M |
| weights | 925 | 1388 | 1851 | 2777 |
| 100 K | 63.3 | 64.1 | 64.5 | 65.5 |
| 200 K | 64.2 | 65.1 | 65.6 | 66.3 |
| 500 K | 64.0 | 65.6 | 65.5 | **66.5** |
| 1 M | 64.6 | 65.5 | 66.0 | 66.3 |
| 2 M | 64.5 | 65.2 | 65.7 | 66.2 |

LVCSR WER [%]

| band | Merger weights / hidden units | | | |
|---|---|---|---|---|
| NNs | 1 M | 1.5 M | 2 M | 3 M |
| weights | 925 | 1388 | 1851 | 2777 |
| 100 K | 39.7 | 39.0 | 38.8 | 38.2 |
| 200 K | 39.0 | 38.5 | 38.3 | **37.7** |
| 500 K | 39.2 | 38.6 | 38.9 | 37.9 |
| 1 M | 39.6 | 38.5 | 38.4 | 38.2 |
| 2 M | 39.7 | 38.9 | 38.4 | 38.2 |

**Table 3.** The Merger Cross-Validation frame accuracies and LVCSR WERs [%]

Cross-Validation frame accuracies [%]

| band NNs | Merger weights | | | |
|---|---|---|---|---|
| weights | 1 M | 1.5 M | 2 M | 3 M |
| 100 K | 65.1 | 66.1 | 66.7 | **67.5** |
| 200 K | 63.9 | 65.2 | 66.0 | 67.0 |

LVCSR WER [%]

| band NNs | Merger weights | | | |
|---|---|---|---|---|
| weights | 1 M | 1.5 M | 2 M | 3 M |
| 100 K | 37.6 | 37.4 | 36.8 | 36.7 |
| 200 K | 38.7 | 37.5 | 37.1 | **36.6** |

hidden units is different for every experiment. The HATS Merger's hidden layer sizes are the same as for TRAP Merger's for 100 K weights in band NNs and roughly half for 200 K weights in band NNs. The HATS Merger cross-validation accuracies and WERs are given in Tab. 3.

The performance of HATS systems is also increasing with increasing size of Merger NN. On the other hand, increasing the size of band classifiers has negative effect on the performance of the whole system - only the architecture accommodating the largest Merger was able to provide comparable performance to a system with smaller band NNs.[2] This shows that the HATS Merger was overloaded by increased number of inputs. Although these inputs carry more information which was able to improve TRAP systems, it cannot be utilized by HATS Merger and, contrary, more inputs seems to bring larger confusion and impair the overall performance.

### 4.1 Detailed Analysis

This section is focused on the band NNs accuracies. Tab. 1 gives the classification accuracies of the $6^{th}$ band NN. The accuracies of band NNs in all bands are shown in Fig. 2. It can be seen, that the classification increases in all bands with increasing NN size, so stagnation in Mergers accuracy cannot be assigned to the degradation of NNs in other bands.

The following analysis was focused on classification accuracy of individual classes by one band NN – $6^{th}$ band was used. The cross-validation data were used for this analysis. The percentage of correctly classified frames per individual class are shown

[2] Further experiments with larger band NNs were not performed as stagnation was observed for TRAP architecture and degradation for HATS.

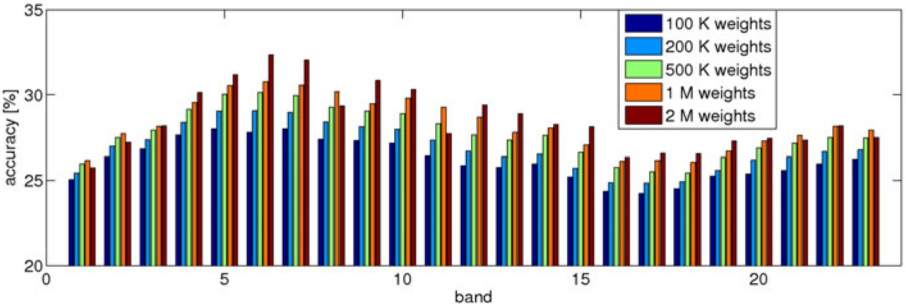


**Fig. 2.** Cross-validation accuracies [%] of all band NNs

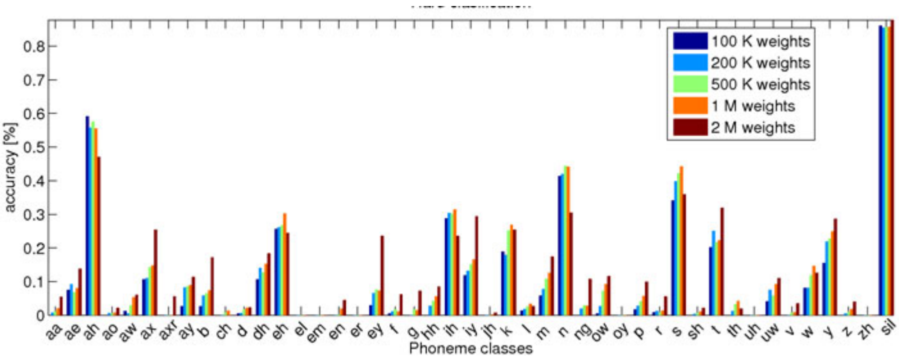


**Fig. 3.** Classification accuracies per individual class [%] – $6^{th}$ band

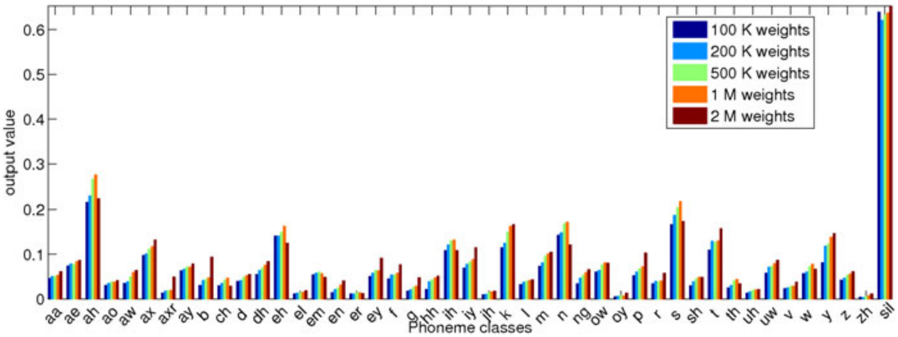


**Fig. 4.** Average value of corresponding NN output for given class [%] – $6^{th}$ band

in Fig. 3. In the next step, the average value of output corresponding to given class was computed over all input vectors labeled as given class, see Fig. 4.

For both analysis can be seen that increased size of band classifier increased classification accuracies/average output value for almost all classes. The exception are band

NNs of size 2M weights which seems to perform significantly better for some classes and have worse performance for others. Overall can be said that the patterns are about the same.

## 5 Conclusions

In this study we have investigated the role of NN size in TRAP/HATS probabilistic feature extraction scheme. This investigation covers both parts of the processing – band NNs and Merger. The band NNs creates some kind of filter which let only particular information to Merger. If the information is lost here, Merger will not be able to achieve high classification accuracy. Thus it is important to use band NNs of such size, which will preserve all necessary information. The Mergers task is to combine particular probability estimates into final ones. It thus has to have enough power to perform this task.

The results obtained on TRAP architecture shows that the system performance increases with both, band NN size and Merger size. But the enlargement of band NNs over some size does not have further positive effect, the performance saturates. The increased size of the merger can compensate for poor band NNs performance to some extent. But the cost in terms of used weights is much higher compared to what is added to band NNs. Over all, it can be said that having more parameters in band NNs does not hurt and leads to good system performance.

Unfortunately, this cannot be said for HATS architecture. Although much better performance was obtained by this architecture when band NNs with 100 K weights were used, increasing size of band NN did not improve the performance. Contrary, degradation was mostly seen and only the architecture with largest merger gained comparable results. Why the improvement seen for TRAP systems is not observed when HATS architecture is used instead? We know, that useful information is contained in activation outputs of larger band NNs, but giving it directly to Merger is not the right way to present it. The HATS system seems to benefit from compact information on Mergers input. From this point of view, the tuning of HATS system in [14] might be questionable, but the authors did not provide the NN sizes to give us a clue where their operation point is. It can be recommended to prefer smaller band NNs when designing the HATS architecture and to validate the designed architecture experimentally.

It would be beneficial to present compact information to the Merger regardless the size of band NNs. Such solution have been already proposed in the form of bottle-neck NN structure [15]. It effectively separates the output size from other parameters of the NN such as number of classes (which is fixed in TRAP architecture) and size of the NN. The possible problem with this approach lies in usage of five-layer NN. It might be difficult to train more complex NN on evolution of just one parameter (a single critical band energy) and also proper setting of sizes of all layers would be more complicated. Nevertheless, this approach seems to be another step in TRAP/HATS feature extraction techniques.

## References

1. Hermansky, H., Ellis, D.P.W., Sharma, S.: Tandem connectionist feature extraction for conventional HMM systems. In: Proc. ICASSP 2000, Turkey (2000)
2. Sharma, S.R.: Multi-stream approach to robust speech recognition, Ph.D. thesis, Oregon Graduate Institute of Science and Technology (October 1999)

3. Hermansky, H., Sharma, S., Jain, P.: Data-derived nonlinear mapping for feature extraction in HMM. In: Proc. Workshop on Automatic Speech Recognition and Understanding, Keystone (December 1999)
4. Athineos, M., Hermansky, H., Ellis, D.P.W.: LP-TRAP: Linear predictive temporal patterns. In: Proc. ICSLP 2004, Jeju Island, KR, pp. 949–952 (October 2004)
5. Tyagi, V., Wellekens, C.: Fepstrum representation of speech signal. In: Proc. of IEEE ASRU, San Juan, Puerto Rico, pp. 44–49 (December 2005)
6. Jain, P., Hermansky, H.: Beyond a single critical-band in TRAP based ASR. In: Proc. Eurospeech 2003, Geneva, Switzerland, pp. 437–440 (2003)
7. Grézl, F., Hermansky, H.: Local averaging and differentiating of spectral plane for TRAP-based ASR. In: Proc. Eurospeech 2003, Geneva, Switzerland (2003)
8. Zhu, Q., Chen, B., Grézl, F., Morgan, N.: Improved MLP structures for data-driven feature extraction for ASR. In: Proc. INTERSPEECH 2005, Lisbon, Portugal (September 2005)
9. Ellis, D., Morgan, N.: Size matters: An empirical study of neural network training for large vocabulary continuous speech recognition. In: Proc. ICASSP 1999, Phoenix, Arizona, USA, pp. 1013–1016 (March 1999)
10. Bourlard, H., Morgan, N.: Connectionist Speech Recognition: A Hybrid Approach. Kluwer International Series in Engineering and Computer Science, vol. 247. Kluwer Academic Publishers, Dordrecht (1994)
11. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press Professional, Inc., San Diego (1990)
12. Hain, T., et al.: The AMI system for the transcription of speech meetings. In: Proc. ICASSP 2007, Honolulu, Hawaii, USA, pp. 357–360 (April 2007)
13. Chen, B., Zhu, Q., Morgan, N.: Learning long-term temporal features in LVCSR using neural networks. In: Proc. ICSLP 2004, Jeju Island, KR (October 2004)
14. Zhu, Q., Stolcke, A., Chen, B., Morgan, N.: Using MLP features in SRI's conversational speech recognition system. In: Proc. INTERSPEECH 2005, Lisbon, Portugal (September 2005)
15. Grézl, F., Karafiát, M., Kontár, S., Černocký, J.: Probabilistic and bottle-neck features for LVCSR of meetings. In: Proc. ICASSP 2007, Honolulu, Hawaii, USA, pp. 757–760 (April 2007)

# Time Dimension in the Dolphin Nick Knowledge Base Using Transparent Intensional Logic

Andrej Gardoň and Aleš Horák

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

**Abstract.** In this paper we describe the analysis and implementation of the time dimension in the Dolphin Nick knowledge base which is based on the theory of the Transparent Intensional Logic (TIL). First, we analyze the basics of the temporal aspect in TIL constructions obtained from natural language sentences and describe its algorithmic form. We present the details of implementation of knowledge base objects corresponding to grammar tenses. Problems in basic time inference using the presented system are discussed with the implemented solutions. Finally we show an example communication with the system regarding the time aspects.

**Keywords:** TIL, Dolphin Nick, knowledge base, time, inference, reasoning.

## 1 Introduction

Processing of the time dimension in computer applications working with natural language (NL) text has been usually simplified in some way – either it has not been necessary to include the time information with common data, or ”ready made” relations have met the goal [8,9]. However, a growing need for ”intelligent” NL processing asks for proper handling of the temporal knowledge contained in NL texts [6]. Standard first-order logic (FOL) tools are not able to ”understand” sentences like (1) in the natural way.

$$\textit{Maria was at home on Sunday.} \tag{1}$$

The relation between ”*Maria was at home*” and ”*Sunday*” can be mapped on a FOL predicate but the information about the time reference is narrowed – *Sunday* is treated as an entity without the interval interpretation. Proper handling of all kinds of temporal information is not possible within the frame of FOL – time-aware theories must be followed [7].

In the following text, we describe the analysis and design of an implemented knowledge based question answering system named Dolphin Nick [2] with regard to the processing of NL time information. The Dolphin Nick system is theoretically based on the higher-order intensional Transparent Intensional Logic (TIL [10]) and the algorithmic model of the knowledge base builds on the idea of a network of constructions in [4, chap. 6].[1] The implementation language of Dolphin Nick is C++ with intensive use of the object paradigm for the TIL modelling.

[1] The motivation behind using TIL as the underlying formalism can be found in [3].

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 323–330, 2011.

## 2 Simple Temporal Aspect in TIL and Dolphin Nick

The meaning of the sentence (1) is in the underlying TIL theory analyzed as a TIL construction (procedure) of a *proposition* with the type of an ”intensional truth-value” $o_{\tau\omega}$. The exact notation resulting from the Normal Translation Algorithm for TIL (NTA [4,5]) is as follows:[2]

$$\lambda w \lambda t [\mathsf{P}_t [\mathsf{Onc}_w \lambda w_1 \lambda t_1 [\mathsf{Does}_{w_1 t_1} Maria [\mathsf{Perf}_{w_1} \mathsf{to\_be\_at\_home}_{w_1}]]]\ \mathsf{Sunday}] \tag{2}$$

Notice that the analysis of verb phrases is based on conclusions presented in [11]. The temporal information in (2) is handled by the terms P (past tense), Onc (frequency *Once*) and Sunday (time interval). The other logical objects (Does, Perf and to_be_at_home) work with the attributes of the verb in the sentence and with the specification of its basic (present tense) construction. The variables $w$ and $t$ express the (possible) world- and time-independence of the respective intension (proposition in this case). Evaluating the truthfulness of such sentence lies in instantiating these variables with a time-moment and a possible world, e.g. the actual one.

The computational form of the TIL construction in the Dolphin Nick system is denoted as DolphinConstruction (DC). Each DC is internally represented with an instance of a C++ class with several attributes including:

- ID – unique identification
- the TIL type of the corresponding TIL construction
- references to all possible Dolphin subconstructions that form the DC (see DCs types below).[3] The network design of the system allows smart handling of the references.
- class inference rules as C++ functions (see [3,1])

In accordance with TIL, there are three types of DCs:

- Variables and trivializations (*simple DCs*, e.g. *Apple*) are represented as mappings between words and C++ objects with unique identification (ID).
- Applications form *compound DCs*. In Dolphin Nick an application is a relation between two DCs of any type.[4] The notation of [DC200 DC190] is used to represent an application in the system's input/output language DOLLY.
- Closures are handled as *stored procedures* – specific code snippets provide fast processing of functions defined using standard lambda calculus.

In the system time information is represented by special subcategory of simple DCs called TObjects. Currently, the system uses two types of TObjects:

- a Continuous interval (CI)
- a Set of Intervals (SI)

[2] To keep things simple, verbal objects are presented as trivializations such as **to_be_at_home**. However, inside the system, the analysis corresponds to the full version as follows from NTA.

[3] As far as the system knows, i.e. stored in the knowledge base as related with the possible world $w_{\text{Dolphin}}$.

[4] Applications with $N$ arguments are expressed as $N$ successive applications of *one* argument.

**Table 1.** Examples of Continuous intervals (CIs)

| CI | START | | DURATION | | Description |
|---|---|---|---|---|---|
| ID | UNIT | VALUE | UNIT | VALUE | |
| 1 | Second | 5 | – | – | the time moment ”5 seconds (after 0)” |
| 2 | Year | 2000 | Week | 1 | the first week of the year 2000 |
| 3 | – | – | – | – | Unspecified (or ”to be specified”) interval. Used in temporal relative sentences. |
| 4 | Stamp | 1.1.2011<br>12:00:00 | Second | 1 | Interval of 1 second since the time specified by a time-stamp string |

**Table 2.** Examples of Sets of intervals (SIs)

| SI ID | CIs | | | | | $\chi$ | Description |
|---|---|---|---|---|---|---|---|
| 20 | 7 | Stamp | 1.4.2011 | Day | 1 | – | Interval ”*Friday and Sunday*” |
| | 8 | Stamp | 3.4.2011 | Day | 1 | | |
| 21 | 9 | Stamp | 7.4.2011 | Day | 1 | Every week | Interval ”*Every Thursday*” |

CI is specified by the size and the relative position to a chosen system zero point[5] both with flexible time units. Table 1 shows some examples of CIs.

SIs are used for representation of discontinuous intervals – such an example interval can be found in the sentence (3) with the corresponding TIL analysis (4).[6]

*Andrej was shopping in Supermarket on (this) Friday and (this) Sunday.* (3)

$$\lambda w \lambda t [\mathbf{P}_t [\mathbf{Thr}_w \lambda w_1 \lambda t_1 [\mathbf{Does}_{w_1 t_1} Andrej \quad [\mathbf{Perf}_{w_1} \mathbf{to\_shop\_in\_supermarket}_{w_1}]]] \mathbf{Friday\_and\_Sunday}] \tag{4}$$

In the sentence the mentioned interval is represented as an SI containing CIs *stamp_dur (1.4.2011,Day,1)* and *stamp_dur (3.4.2011, Day, 1)*. Generally, an SI is a set of CIs with its own ID and (possibly) a characteristic functions $\chi$. Internally, each characteristic function is implemented as a C++ function providing the appropriate recurrent time interval with the ability to check particular dates for interval membership. Table 2 shows some example SIs. CIs and SIs together allow to capture the general definition of time intervals in TIL as *classes of time-moments*, $(o\tau)$-objects.

Now, after the definition of TObjects, we are ready to describe the implementation of time adverbs which are essential for processing time related sentences. In general, time adverbs are DCs that according to a proposition construct DCs of type $(o)(o)\tau$, i.e. sets of TIL time intervals. As mentioned above such set can be represented by an SI object.

In the construction (4) the subconstruction $[\mathbf{Thr}_w \lambda w_1 \lambda t_1 [\mathbf{Does}_{w_1 t_1}]] \ldots$ denotes SI 20 that represents all continuous intervals in which the respective proposition was true. It is possible to request a check of membership of CI or other SI to SI 20 for the purpose

[5] e.g. the year 0 in the Gregorian calendar.

[6] Although the DolphinNick system uses the DOLLY language to represent an analysis of a sentence, in the following examples the NTA output is presented for readability.

of questions like *"Was Andrej shopping on this Saturday?"*. An internal code reveals that *Saturday* is not a member of SI 20 resulting to the answer *"No"*.

## 3 Actions and Verbs

An episodic verb in a sentence introduces an action. For example sentence (5) is talking about Maria reading a sentence.

$$\textit{Maria is reading: "This day is nice."} \tag{5}$$

To capture the dynamic behavior of such actions, TIL uses the notions of Events and Episodes [11,4]. For illustration, an event represents a "description of a snapshot" in a specific time moment of the action and an episode is represented as a class of such events. An event can be described as a conjunction of propositions: *It is time $T_2$ and Maria is reading word "day" and before $T_2$–$T_1$ seconds Maria was reading word "this" and after $T_3$–$T_2$ seconds Maria will read word "is"*... Declaratively each event describes the whole action from the point of view of the given time moment.

Dolphin Nick implements TIL episodes as information connected with propositions *"Maria reads word "this" in interval $I_1$... Maria reads the sentence: "This day is nice" in $I_5$"*. Such information is internally represented as a DC episode (DCE) with the type $(o)\pi$[7], i.e. a set of special propositions (DCEP). The running time of a DCE is defined within the proposition *"Maria reads the sentence... "* which must be defined in all time moments from the reference interval. Such a proposition is called basic [4, p. 65] and it is essential for every single DC episode.

Any DCE is identified by a verb in the input sentence. Actually, a verb in TIL is an object of type $(o(o\pi)(o\pi))_w$ which represents a relation between an upshot and a labour episode. As a result every verb defines (connects) two independent basic propositions for labour and upshot episode. However, in many cases the two episodes coincide[8] and are thus represented by one basic proposition.

DCE from the sentence (5) contains all the propositions (DCEPs) derived from the basic proposition denoted by the whole sentence – *"Maria is reading: This day is nice"*. The DCEP derivations consists in time specification of the original proposition ("*(5) in interval $I_1$*") and its (sub)events ("*Maria reads word...*"), which are derived by the internal logic of the used verb (see Figure 1). All derived DCEPs (children) form an extensional subclass of the input proposition (mother) for the given interval and possible world.

The mother proposition has the information how many (children) exist and internal processes of the system ensure consistence between mother and children DCs bidirectionally.

Within the learning phase, the proposition depicted by the construction [**Does** ...] is responsible for the creation of the respective DCE. Declaring this proposition as *true* in the current time $t_1$ and the possible world $w_{\text{Dolphin}}$ means that there exists an episode in the knowledge base that occurs in $w_{\text{Dolphin}}$, its running time includes the time

[7] The $\pi$ type is a short for $o_{\tau\omega}$, the type of propositions.

[8] e.g. for *singing* the verb's process and the result are identical.

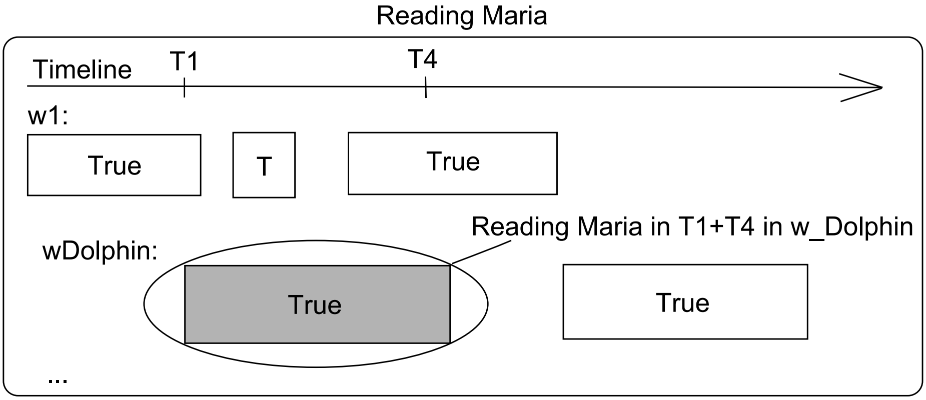


**Fig. 1.** DCEP for the sentence *Maria is reading "This day is nice"* in $T_1$–$T_4$

moment $t_1$ and its protagonist is Maria. The internal inference mechanism [3] creates the appropriate DCEP, inserts it to the new DCE and adds this episode do the set of episodes connected with the reading action.

## 4 Time Inference and Grammatical Tenses

Let us now look at the following sentence (6) and the constructions (10) and (11)[9] that correspond to the two expanded sub-readings of (6), i.e. (7) and (8). The frequency adverb object **Onc** provides a set of intervals for the proposition denoted by the construction (9). These intervals represent situations in which the respective action happened but exact time specifications are not known.

*Maria was at home on Tuesday but not on Wednesday.* (6)

*Maria was at home at least once during Tuesday.* (7)

*Maria was not at home during the whole Wednesday.* (8)

$$\lambda w_1 \lambda t_1 [\mathbf{Does}_{w_1 t_1} \mathbf{Maria} [\mathbf{Perf}_{w_1} \mathbf{to\_be\_at\_home}_{w_1}]] \tag{9}$$

$$\mathbf{True} :: \lambda w \lambda t [\mathbf{P}_t [\mathbf{Onc}_w (9)] \mathbf{Tuesday}]] w_{\text{Dolphin}}] t_1] \tag{10}$$

$$\mathbf{False} :: \lambda w \lambda t [\mathbf{P}_t [\mathbf{Thr}_w (9)] \mathbf{Wednesday}]] w_{\text{Dolphin}}] t_1] \tag{11}$$

The sentence (6) says that Maria was at home at least once during Tuesday (for an unknown period of time) and that she was not at home during the whole Wednesday.

The current Dolphin Nick system uses the approach based on the interval inference to process the sentence (6). The proposition (9) keeps the knowledge about existence of all sets obtained through the respective time and frequency adverbs, such as **Onc** or **Thr**. This knowledge is used by the Time inference to process the above mentioned

[9] To keep readability the time intervals use NL representations.

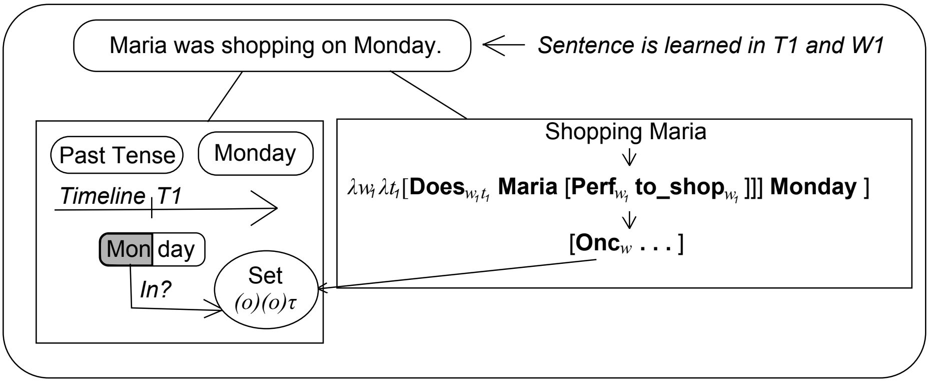

Maria was shopping on Monday.
Sentence is learned in T1 and W1
Past Tense
Monday
Timeline T1
Mon day
In?
Set
(o)(o)τ
Shopping Maria
λw₁λt₁[Does w₁t₁ Maria [Perf w₁ to_shop w₁ ]]] Monday ]
[Onc w . . . ]

**Fig. 2.** Processing a sentence with the past tense

sentences and assure consistency in the sense that incompatible intervals for an already learned proposition are rejected. Thus if we say (after (6)) that

$$\textit{Maria was not at home during the whole Tuesday.} \qquad (12)$$

then logically one of the sentences (6) and (12) must be false and the contradiction is revealed by the system and the new knowledge is rejected.[10]

Finally, the implementation of grammatical tenses in the system is presented. In the case of Past, Perfect and Future tenses, their task lies in the declaration of a time interval that is checked against the time interval set identified by the time adverb. Figure 2 illustrates the situation. The Past tense object $P$ selects a subset of the interval identified by "*Monday*" according to the time moment $T_1$. This subset is than included into the set constructed by $[\textbf{Onc}_w \ldots]$. If the grammatical tense includes the progressive form of the verb, then we declare a time interval in which the respective verb's action is taking place and an appropriate episode saying that the sentence is true throughout this interval is created.

## 5 Example Questions and Answers

To sum up the previous text, we want to demonstrate the time-related capabilities of the Dolphin Nick system on a series of questions and answers from the system. For the purpose of readability, the sentences are displayed in natural language, although the system currently uses the DOLLY language for both input and output. The prompt "U:" precedes the user input and "DN:" the system's answer.

1. U: *G.W. Bush was the president of the USA from January 20th 2001 to January 20th 2009.*

[10] Any new information is put into the part of the system Memory called Cache. During the processing of Cache the previously stored knowledge is consulted and in the case of a contradiction the information from the Cache is erased. If the processing succeeds the input is learned into the main Memory.

U: *Who was the president of the USA on January 1st 2009?*
DN: *G.W.Bush*

2. U: *Power cables were installed at the Japan plant on March 22nd 2011.*
U: *Were power cables were installed on March 22nd 2011 14:00?*
DN: *I do not know but Power cables were installed at the Japan plant on March 22nd 2011.*

3. U: *Christopher Columbus was born on October 31st 1451.*
U: *America was discovered on October 12th 1492.*
U: *George Washington was the first president of USA in 1789–1797.*
U: *Was America discovered before Christopher Columbus was born?*
DN: *No. Christopher Columbus was born on October 31st 1451 and America was discovered on October 12th 1492.*
U: *Was Columbus born after George Washington was the president of USA?*
DN: *No. Christopher Columbus was born on October 31st 1451 and George Washington was the first president of USA in 1789–1797.*

4. U: *Dow Jones was at 12025 on Tuesday 12:00.*
U: *Dow Jones was at 12024 during the whole Tuesday.*
DN: *Cannot learn last sentence, since Dow Jones was at 12025 on Tuesday 12:00.*

5. U: *Maria was reading when Andrej came.*
U: *Has ever Maria been reading?*
DN: *Yes.*
U: *When was Maria reading?*
DN: *When Andrej came.*

The last group of sentences introduces relative time intervals which have not been fully described in this text. Briefly, the proposition denoted by *"Maria was reading"* has the *True* value in the interval $I_1$ that is defined by the proposition *"When Andrej came"*. This second interval is a singleton represented by a special (variable) time moment. The first question was answered positively since the system identified $I_1$ as a time interval in which Maria was reading. The system Memory then provides the corresponding DC which defines the interval $I_1$ as an answer to the second question.

## 6 Conclusions and Future Work

We have presented the implementation details of the temporal aspect in the Dolphin Nick knowledge base system. The computer model of processing of verbs and actions in natural language texts is based on the theory of the Transparent intensional logic (TIL). For Dolphin Nick, two types of TObjects – a Continuous Interval (CI) and a Set of intervals (SI) were introduced to store the time related information together with the system processes behind verbs, grammatical tenses, time adverbs and episodes that were described in examples.

All examples mentioned in the text are fully supported by the present version of the Dolphin Nick system. Our future research is directed towards deduction and processing of sentences with implication modifier.

**Acknowledgments.** This work has been partly supported by the Czech Science Foundation under the project P401/10/0792 and by EC FP7 project ICT-248307.

## References

1. Gardoň, A.: The Dolphin Nick Project (2011), http://www.dolphin-nick.com/
2. Gardoň, A., Horák, A.: The Learning and Question Answering Modes in the Dolphin System for the Transparent Intensional Logic. In: Proceedings of the Recent Advances in Slavonic Natural Language Processing (RASLAN 2007). Czech Republic (2007)
3. Gardoň, A., Horák, A.: Knowledge Base for Transparent Intensional Logic and Its Use in Automated Daily News Retrieval and Answering Machine. In: 3rd International Conference on Machine Learning and Computing (ICMLC 2011), vol. 1, pp. 59–63. IEEE, Singapore (2011)
4. Horák, A.: The Normal Translation Algorithm in Transparent Intensional Logic for Czech. Ph.D. thesis, Masaryk University (2002)
5. Horák, A., Kadlec, V.: New Meta-grammar Constructs in Czech Language Parser synt. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 85–92. Springer, Heidelberg (2005)
6. Jackson, P., Moulinier, I.: Natural language processing for online applications: Text retrieval, extraction and categorization. John Benjamins Pub. Co., Amsterdam (2007)
7. Kröger, F., Merz, S.: Temporal logic and state systems. Springer-Verlag New York Inc., Heidelberg (2008)
8. Liu, H., Singh, P.: ConceptNet–a practical commonsense reasoning tool-kit. BT technology journal 22(4), 211–226 (2004)
9. Moldovan, D., Clark, C., Harabagiu, S.: Temporal context representation and reasoning. In: International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, pp. 1099–1104 (2005)
10. Tichý, P.: Collected Papers in Logic and Philosophy. Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press (2004)
11. Tichý, P.: The semantics of episodic verbs. Theoretical Linguistics 7, 264–296 (1980)

# Towards Automatic Annotation of Sign Language Dictionary Corpora[*]

Marek Hrúz, Zdeněk Krňoul, Pavel Campr, and Luděk Müller

Department of Cybernetics
University of West Bohemia
306 14, Plzen, Czech Republic
{mhruz,zdkrnoul,campr,muller}@kky.zcu.cz

**Abstract.** This paper deals with novel automatic categorization of signs used in sign language dictionaries. The categorization provides additional information about lexical signs interpreted in the form of video files. We design a new method for automatic parameterization of these video files and categorization of the signs from extracted information. The method incorporates advanced image processing for detection and tracking of hands and head of signing character in the input image sequences. For tracking of hands we developed an algorithm based on object detection and discriminative probability models. For the tracking of head we use active appearance model. This method is a very powerful for detection and tracking of human face. We specify feasible conditions of the model enabling to use the extracted parameters for basic categorization of the non-manual component. We introduce an experiment with the automatic categorization determining symmetry, location and contact of hands, shape of mouth, close eyes and others. The result of experiment is primary the categorization of more than 200 signs and discussion of problems and next extension.

## 1 Introduction

Sign language (SL) is a communication form mainly used by deaf or hearing impaired people. In this language visually transmitted manual (MC) and non-manual (NMC) components are used to convey meaning. The MC consists of hand shape, palm orientation and the arm movement. The NMC component consists of face expression, body pose and lip movement. Because the majority language (usually the language used by the hearing) is the secondary language of the Deaf a bi-directional translation is highly important for better Deaf orientation in our day-to-day shared social environment. Currently, human interpreters provide this translation but their service can be expensive and not always available. Therefor systems of SL recognition and synthesis are being developed [1,2,3]. The results from these fields of research can be used in various ways.

[*] This research was supported by the Grant Agency of Academy of Sciences of the Czech Republic, project No. 1ET101470416 and project No. GAČR 102/09/P609, by the Ministry of Education of the Czech Republic, project No. ME08106 and by the grant of the University of West Bohemia, project No. SGS-2010-054.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 331–339, 2011.

Our goal is to use the recognized features for an automatic categorization of signs for the use in a SL dictionary. The proposed categorization algorithm considers sign categories corresponding to the entries in the symbolic notation HamNoSys (HNS) and SignWriting (SW)[1]. Symbolic notations are used to describe the sign. Usually these notations are created manually which is very time consuming. This process is influenced by the skills and experience of the human annotators. On the other hand automatic categorization of video files is deterministic provided the same input parameters. Also it fastens the work of human annotators who will only need to correct the mistakes of the automatic annotation. This annotation allows to search among the signs and enables the translation from SL to spoken language.

## 2 Related Work

MC recognition is closely related to tracking. There are many methods that vary depending on the scenario. Sometimes markers or color cues are used to help the process. A good survey can be found in [4]. We also refer to [5]. Our approach is based on color segmentation and object detection and description. Similar approach can be found in [6]. In our work we experiment with linear dimension reduction methods to obtain better tracking results.

For NMC signal, there are generative parametric models commonly used to track and synthesize faces in images and video sequences. We can distinguish two types of automatic face tracking algorithms. The first type is feature-based, matching the local interest points between subsequent frames, such as a 3D pose tracker [7] and 3D deformable face tracking [8]. The second type is appearance-based, using generative linear models of face appearance. There are Active Appearance Models (AAM) [9] and 3D Morphable Models [10].

AAM is a combined model of shape and texture. It ensures precise alignment, is very powerful and efficient to describe the movements in the face. The original proposal is used for identification of different faces as well as tracking [9]. Further improvements of AAM for local inter-frame appearance constraint optimization are integrated [11]. There are 3D AAM including 3D models to cover non-linear changes in the observed data. In most cases, the condition is pre-aligned data points arranged in the training images.

## 3 Data

Data for our experiment are selected signs from the on-line dictionary [12]. The dataset consists of pairs of synchronized video files capturing one speaker from two different views in the same lighting conditions. The recordings of the first and second view are RGB color images in HD resolution, 25 frames per second with high-quality compression. The first view captures the entire body of the speaker and the second one is a detail of the face, see Fig. 1. Audio track is not included. Totally 213 signs (video files) are considered.

[1] www.sign-lang.uni-hamburg.de/projects/hamnosys.html, www.signwriting.org/

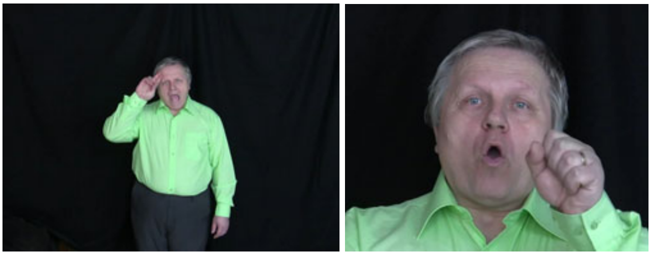


**Fig. 1.** Example of considered data. Left - manual component, right - non-manual component.

# 4 Hand Tracking

The hand tracking is based on skin color segmentation and object description. Because of the nature of our data we can assume constant lighting and environment conditions. This makes the problem of tracking much easier but one has to still account for the occlusions and self-occlusions occurring in SL (for example see [13,14]).

## 4.1 Skin Color Segmentation

Because we are working with data of a SL dictionary we can assume that there will be not many performers and the characteristics of the video data will be constant or at least piecewise constants. That is why we use a constant skin color model in a form of a look-up-table. There have been a lot of papers published in this field. The approaches usually differ in the color models used for skin color representation and a parametric or non-parametric description of the model. We work with the native RGB color space and a hybrid model which in the end yields a non-parametric model in the form of a look-up-table. We train a Gaussian Mixture Model (GMM) from examples of skin color manually selected from our database. We threshold and scale the probability of the model to obtain a $256 \times 256 \times 256$ look-up-table with values from 0 to 255. We were inspired by the work [15] which we refer to for further details.

## 4.2 Tracking

In the scenario of SL movements tracking there are several objects of interest. The head that is usually static but changes a lot in the appearance. The hands that move rapidly, change the shape and orientation to the camera. We assume that the changes of the appearance occur slowly relatively to the camera frame rate. This should enable us to track the objects in a discriminative manner.

We define a tracker for each object we want to track. In our case there are three trackers. The tracker contains several discriminative models for object tracking. The number of models depends on the number of events we want to take into account. In our case there are 4 models. Model of non-occluded object, model of the change of state from non-occluded to occluded, model of occlusion and model of the change of state from occluded to non-occluded. This is due to big changes in the appearance of the objects when they travel from one state to the other. Each model is a 4 mixture GMM in a 5D space that is defined by the properties of the objects. In the first frame the trackers are initialized by the object lying in a predefined region based on the knowledge of the starting pose of a performer. In the next frame a new set of objects is detected. Each tracker compares every object with the identified object from the last frame via the discriminative probability model. The input vector for the model is a 5D vector of relative differences. Then the probability of this vector is the probability of the unknown object to be the tracked one. This probability will be noted as $p_m(t_i|o_j)$, where $m$ is a specific model, $t_i$ is the $i^{th}$ tracker and $o_j$ is the $j^{th}$ object.

**Object Description.** In the last paragraph we mentioned a 5D vector of relative differences that is used for object comparison. The vector is obtained as follows:

1. Use skin color segmentation to obtain all possible body parts.
2. Eliminate all segments that do not fulfill the defined conditions (size, width/ height ratio)
3. For each object compute - bounding box, Hu moments of the contour, area of the object and perimeter of the contour
4. From the information in point 3. compute for every tracker/object pair - normalized correlation between object image and tracked object image, normalized distance between their contours (computed from Hu moments), relative difference between their bounding box areas, relative difference between their perimeters, relative difference between their areas, relative difference between their velocity and location

This procedure yields a 7D vector for each tracker/object pair. Next, we want to find a transformation that reduces the correlation between the features and possibly reduces the dimensionality of the model. We experimented with Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA) and Heteroscedastic LDA (HLDA). These experiments are out of the scope of this paper but note that HLDA transformation into a 5D feature space resulted in the best performance.

**Configuration Determination.** Next, we want to determine which object belongs to which tracker. Let $\mathcal{O} = \{o_j\}, j = 1..N$ where $N$ is the number of objects be a set of detected body parts. Let $\mathcal{T} = \{t_i\}, i = 1..3$ be a set of trackers. A configuration $\mathcal{C}$ is a mapping $\mathcal{T} \rightarrow \mathcal{O}$ that fully describes which tracker tracks which object. In SL scenario there exist 5 cases depicted in Figure 2 that describe the mutual relation between body parts. Each case is conditioned by the number of body parts detected. This enables us to hypothesize only about the plausible configurations. The algorithm for configuration determination is as follows:

1. Based on the number of detected body parts select a $C_k \in \mathcal{C}$ that fulfills the condition
2. Compute the log-likelihood of the selected configuration

$$L_k = \sum_{i=1}^{3} \log p_m(t_i|C_k(t_i)) \qquad (1)$$

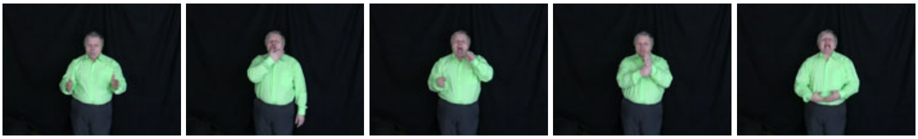


**Fig. 2.** Five possible cases of hand/head mutual relation. Note that several configurations may represent each case. This is due to the fact, that the case does not tell us which object is which.

3. If this is the maximum likelihood seen so far, store it as a new maximum
4. If there are no more configurations to test, select the $C_k$ with maximum $L_k$ as the recognized configuration, else go to point 1

In Eq. 1 we have to select a proper model $m$ to evaluate the probability. The model is determined by the configuration $C_k$. Based on the last known configuration we are able to tell from which state the individual objects travel to the hypothesized state defined by $C_k$. The model can be different for each tracker $i$.

## 5 Head Tracking

The NMC integrates several face expressions, mouthing, 3D position of head. To ensure robust processing, we assume the multi-resolution combined active appearance model (AAM) [9]. The AAM traces the position and local shape of the face by a combination of two linear models for shape and texture.

The training set consists of 51 images selected from the dataset. The training set includes NMC of complex face gestures and head position. Naturally NMC involves 3 DOF for rotations (the $x$, $y$ image axes and the $z$ optical axis). Nevertheless, rotation around the image axes (pitch and yaw) has to be incorporated to the shape model because we consider 2D AAM. Rotation around the optical axis (roll) is described by the pose parameters and is not incorporated in the shape model. This is done by manual normalization of the training frames to get the outer lip and the eye corners horizontal. Thereafter the PCA produces the basic shape $s_0$ plus linear combination of $N$ shape eigenvectors $s_i$:

$$s = s_0 + \sum_{i=1}^{N} F_i s_i. \tag{2}$$

The first 9 principal components preserve $97.5\%$ of variance. Illustration of the shape parameters $F_1..F_4$ is in Fig. 3.

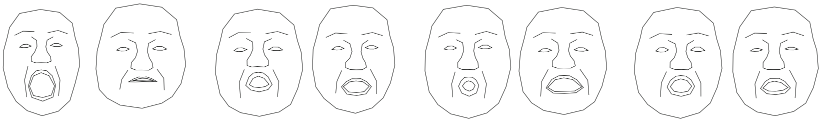


**Fig. 3.** First four modes of the shape model for $\pm 150\%$ of standard deviation

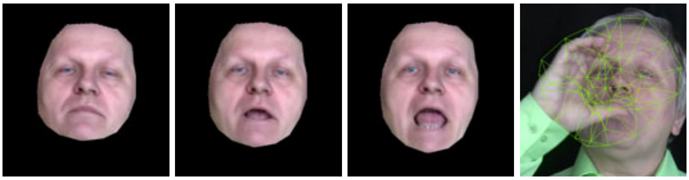


**Fig. 4.** Final fitting of AAM, from the left: fitted appearance of three consecutive input frames and incorrect tracking caused the occlusion

The appearance of AAM is an image defined as the RGB intensity. The eigenvectors are obtained by second PCA on warped training images. $41$ texture parameters describe $97.5\%$ of variance. Finally, combined AAM operates with a single set of parameters $c$ to get the best fit of the AAM in an input video frame. Vector $c$ is obtained by another PCA computed from the appropriately weighted shape a texture parameters.

The illustration of head tracking is in Fig. 4. AAM is sensitive to the initial shape and can end in local minima. Therefore the searching algorithm requires initial localization of face in the first frame of each processed video file. The most likely area showing the speaker's face is detected via a tree-based 20x20 gentle adaboost frontal face detector [16].

## 6 Experiment

The aim of the experiment is to prove the potential of automatic categorization of lexical signs. In the experiments we make use of parameters from tracking. From the tracking of MC we have obtained a contour of hands and head. The values of the contour are in absolute image coordinates. That means that position is also encoded into the contours. From the tracking of NMC we have obtained the shape and texture parameters, Sec. 5. The categories for MC and NMC were chosen similar to the linguistic categories of signs. The linguists have not yet established a universal categorization of signs so we tried to choose more abstract categories. This approach will allow us to describe more detailed categories by combination.

**Categorization of Manual Component.** For this experiment we have chosen categories summarized in Table 1.

To determine the category of a sign we need to compute 2D trajectories of the centroids of the contours. Then the sum of variance of $x$ and $y$ components of the trajectory

**Table 1.** The sign categories chosen for the experiment

| *Hand movement* | *Body contact* | *Hand location* | *Head* |
|---|---|---|---|
| one handed | no contact | at waist | mouth wide open |
| two handed | contact of head and right hand | at chest | mouth wide closed |
| symmetric | contact of head and left hand | at head | lip pressed together |
| non-symmetric | contact of hands | above head | lip pucker |
| | contact of everything | | closed eyes |

determines whether the sign is one handed or two handed. If the variance is sufficient enough it means the hand has moved. To determine the symmetry of the trajectory we compute the sum of absolute values of Pearsons correlation coefficients for $x$ and $y$ positions of both hands. If the trajectories are correlated enough (better than 0.89 each dimension) we claim the sign trajectories are symmetric. The absolute value of the correlation coefficient reflexes the anti-symmetry that occurs in symmetric signs. The location of hand symbolizes what space relative to the location of the head has the hand occupied the most. We compute a histogram of relative $y$ positions of hands consisting of 5 bins. The bins are chosen so that they correlate with the categories. Then the category connected to the most occupied bin is chosen. This approach can fail if the sign duration is relatively small to the video duration. That is why we consider only the segment of the video where the hands are moving and are out of starting position. The last category is body segment contact. For now we can only tell whether the objects occlude each other relative to the camera or touch each other. This is a necessary condition for the body parts contact, but not sufficient. Further experiments are needed. This condition is met when two trackers report the same object as the tracked one. This can be recognized easily since both (or all three) body parts will be represented by the same contour.

**Categorization of Non-manual Component.** In this experiment, we focus on shape information. The information includes the position of face in an image extracted from positional parameters and geometric information extracted from the shape model.

We consider categories of NMC uniquely described by a predetermined subset of all parameters. Few of the categories are mentioned in Table 1. The parameters $X$ describing one category tend to cluster around their single mean value. We considers a simple Gaussian model as the univariate normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$.

The minimum variance unbiased estimator provides us with the sample mean ($\hat{\mu}$) and variance ($\hat{\sigma}$) for randomly selected signs that are manually labeled to the categories. We consider the likelihood of parameterized frame and the category $\mathcal{C}$ as:

$$L(\mathbf{x}, \mathcal{C}) \approx \prod_{i \in \mathcal{C}} f_{\mathcal{C}}(x_i | \hat{\mu}, \hat{\sigma}^2). \tag{3}$$

The algorithm determines (3) for all categories and all frames of the video file. The likelihood of the category $\mathcal{C}$ given the sign $\mathcal{S}$ (the video file) can be derived from the maxima over all frames:

$$L(\mathcal{C}|\mathcal{S}) = \max_{\forall \mathbf{x}} L(\mathbf{x}, \mathcal{C}) \tag{4}$$

The positional parameters describe the $x, y$ translations and rotation in the optical axis independently. However, the contribution of parameters of the shape model into the particular categories in consequence of the used PCA is not evident. Albeit, for example, the first shape parameter describes the opening of the mouth very well , see Fig. 3, however the remaining shape parameters incorporate the partial opening of the mouth as well.

For robust fit of (3), we use the shape parameters only for back projection to the shape $s$ (2). Normalization of the training set of AAM ensures that $s$ is always horizontally aligned. This condition enables a definition of a new set of derived parameters: height, width of lips, closed eyes and raised eyebrows. A category ”small lip rounding”, for example, incorporates two derived parameters related to width and height of the lip.

## 7 Conclusion

The MC tracking algorithm has a 94.45% success rate. This rate was computed against manually annotated video files. A sign was tracked successfully if in every frame the configuration was determined according to the annotation.

The proposed face tracking algorithm fails if the hands occlude significant parts of the face (eyes, nose or mouth), see Fig. 4 on right. The tracking was successful approximately in 95% of signs.

In general, an automatic categorization provides additional information about lexical signs and extends the potential of searching. In the experiment, we consider only a subset of sign categories that can be automatically derived from the features from tracking. These categories can be expressed in a writing form as well, for example by the symbolic notations HNS and SW [17]. The user of the on-line dictionary can search for signs using one of the notation systems and form new search request entering relevant symbols. For every sign we are able to determine the confidence factor for every defined category.

Tracking results provide additional information about the sign. However, for example, repetitive movements of head or hand shape categorization require more complex models. Other categories such as nose folding, forehead wrinkles, cheeks inflate, presence of tongue and teeth require further research in particular with the texture parameters.

## References

1. Aran, O., Burger, T., Caplier, A., Akarun, L.: Sequential belief-based fusion of manual and non-manual information for recognizing isolated signs. In: Sales Dias, M., Gibet, S., Wanderley, M.M., Bastos, R. (eds.) GW 2007. LNCS (LNAI), vol. 5085, pp. 134–144. Springer, Heidelberg (2009)
2. Trmal, J., Hrúz, M., Zelinka, J., Campr, P., Müller, L.: Feature space transforms for czech sign-language recognition. In: Interspeech 2008, pp. 2036–2039 (2008)
3. Krňoul, Z., Kanis, J., Železný, M., Müller, L.: Czech text-to-sign speech synthesizer. In: Popescu-Belis, A., Renals, S., Bourlard, H. (eds.) MLMI 2007. LNCS, vol. 4892, pp. 180–191. Springer, Heidelberg (2008)
4. Ong, S., Ranganath, S.: Automatic sign language analysis: A survey and the future beyond lexical meaning, pp. 873–891 (2005)
5. Zieren, J., Canzler, U., Bauer, B., Kraiss, K.: Sign Language Recognition, Advanced Man-Machine Interaction - Fundamentals and Implementation, pp. 95–139 (2006)
6. Hrúz, M., Campr, P., Železný, M.: Semi-automatic annotation of sign language corpora (2008)
7. Wang, Q., Zhang, W., Tang, X., Shum, H.Y.: Real-time bayesian 3-d pose tracking. IEEE Transactions on Circuits and Systems for Video Technology 16(12), 1533–1541 (2006)

8. Zhang, W., Wang, Q., Tang, X.: Real time feature based 3-d deformable face tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 720–732. Springer, Heidelberg (2008)
9. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 681–685 (2001)
10. Volker, B.: Face recognition based on a 3d morphable model. In: Proceedings of FGR 2006, pp. 617–624. IEEE Computer Society, Washington, DC, USA (2006)
11. Zhou, M., Liang, L., Sun, J., Wang, Y.: Aam based face tracking with temporal matching and face segmentation, pp. 701–708 (2010)
12. Campr, P., Hrúz, M., Langer, J., Kanis, J., Železný, M., Müller, L.: Towards czech online sign language dictionary - technological overview and data collection, Valletta, Malta, pp. 41–44 (2010)
13. Piater, J., Hoyouyx, T., Du, W.: Video analysis for continuous sign language recognition. In: LREC 2010, 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (2010)
14. Buehler, P., Everingham, M., Zisserman, A.: Employing signed tv broadcasts for automated learning of british sign language. In: LREC 2010, 4th Workshop on the Representation and Processing of Sign Languages (2010)
15. Aran, O., Ari, I., Campr, P., Hrúz, M., Kahramaner, D., Parlak, S.: Speech and sliding text aided sign retrieval from hearing impaired sign news videos, Louvain-la-Neuve, TELE, Universite catholique de Louvain, pp. 37–49 (2007)
16. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR 2001, 4th Workshop on the Representation and Processing of Sign Languages, IEEE Computer Society Conference (2001)
17. Krňoul, Z.: New features in synthesis of sign language addressing non-manual component. In: LREC 2010, 4th Workshop on the Representation and Processing of Sign Languages, ELRA (2010)

# Unsupervised Topic-Oriented Keyphrase Extraction and Its Application to Croatian

Josip Saratlija, Jan Šnajder, and Bojana Dalbelo Bašić

Faculty of Electrical Engineering and Computing,
University of Zagreb, Croatia
{josip.saratlija,jan.snajder,bojana.dalbelo}@fer.hr

**Abstract.** Labeling documents with keyphrases is a tedious and expensive task. Most approaches to automatic keyphrases extraction rely on supervised learning and require manually labeled training data. In this paper we propose a fully unsupervised keyphrase extraction method, differing from the usual generic keyphrase extractor in the manner the keyphrases are formed. Our method begins by building topically related word clusters from which document keywords are selected, and then expands the selected keywords into syntactically valid keyphrases. We evaluate our approach on a Croatian document collection annotated by eight human experts, taking into account the high subjectivity of the keyphrase extraction task. The performance of the proposed method reaches up to $F1 = 44.5\%$, which is outperformed by human annotators, but comparable to a supervised approach.

**Keywords:** Information extraction, keyphrase extraction, unsupervised learning, k-means, Croatian language.

## 1 Introduction

Keyphrase extraction is an efficient method for document summarization that improves document retrieval. There are two approaches for labeling documents with keyphrases: *keyphrase assignment*, which labels documents with phrases from a predetermined vocabulary, and *keyphrase extraction*, which labels documents with phrases extracted from document text. Both approaches usually apply supervised techniques that rely on manually labeled training data, which is usually expensive or difficult to obtain. Unsupervised techniques require no labeled data, but rarely achieve the performance of supervised methods.

In this paper we present an unsupervised method for keyphrase extraction. Most keyphrase extraction approaches apply a generic architecture commonly referred to as the *generic keyphrase extractor* [7,8]. This approach consists of two main phases: *candidate phrase generation*, which extracts a set of phrases from a document, and *phrase scoring and selection*, which ranks the extracted phrases and selects a set of keyphrases. Phrase score is usually calculated based on the score of the constituent words. Our approach differs from the generic keyphrase extractor in that keyword scores are not used for keyphrase selection, but rather for forming of keyphrases. We first extract the keywords based on the score obtained by clustering of topically related words

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 340–347, 2011.

and then expand the extracted words into phrases. We perform a thorough evaluation of the proposed method on an expert-annotated collection of documents in Croatian language.

The rest of the paper is organized as follows. In the following section we describe some of the related work. Section 3 describes our keyphrase extraction method. In Section 4 we describe the evaluation methodology and discuss the experimental results. Section 5 concludes the paper and outlines future work.

## 2 Related Work

There are many different techniques that have been applied in document labeling and text categorization. Supervised approaches usually apply machine learning algorithms on a labeled corpus to build a classifier used mainly in the phrase scoring and selection phase. Some of applied machine learning algorithms include: C4.5 decision tree induction [17,5], Naïve Bayes classifier [14,6,1], and support vector machine [20,9]. Supervised approaches usually outperform unsupervised methods, but the cost of data labeling imposes a significant problem.

Semi-supervised techniques try to alleviate the data labeling problem while maintaining the performance of supervised methods. This is usually done by introducing the unlabeled data to boost the learning accuracy acquired on sparse labeled data [10,3].

Most unsupervised approaches use the tf-idf weighting scheme, which has shown to be sufficiently effective for keyword extraction [6,11]. Many unsupervised approaches use tf-idf as a baseline and refine it using additional linguistic knowledge such as part-of-speech [11]. Approaches proposed in [12,4] use clustering algorithms to group together the semantically related words, which are then used for keyword extraction or keyphrase scoring. A comprehensive overview and a systematic evaluation of unsupervised keyphrase extraction approaches is given in [8].

As regards the keyphrase extraction for Croatian language, two approaches have been published thus far: a supervised Naïve Bayes approach [1] and an unsupervised approach based on tf-idf scoring [15].

## 3 Topic-Oriented Keyphrase Extraction

Our approach consists of two separate phases: (1) *keyword extraction*, which extracts keywords from documents, and (2) *keyword-to-keyphrase expansion*, which expands the found keywords into phrases.

### 3.1 Keyword Extraction

The aim of this phase is to identify words in the documents that are most likely to be keywords. This phase is based on the approach proposed in [4]. It consists of three sequential steps: (1) *word clustering*, (2) *tagging documents with clusters*, and (3) *keyword selection*.

**Word Clustering.** This step starts by mapping the words to vectors where each component corresponds to a certain document with the value being equal to the number of occurrences of that word in the document. After the words have been mapped into the vector space, they are clustered with $k$-means algorithm [13] using $k$-means++ centroid initialization [2] and the cosine measure as the similarity measure. In [4] these clusters are said to be semantically related, but our experiments have shown that they are not necessarily semantically but rather topically related. The main difference is that topical relatedness is not corpus-invariant: if a different corpus were to be used, different relations would be found depending on the topics covered in the corpus.

**Tagging Documents with Clusters.** In this step each document is tagged with a single word cluster that achieves the greatest document-cluster score $S\left(D,C\right)$ defined as:

$$S\left(D,C\right)=\sum_{w\in D\cap C}S\left(w,D,C\right) \tag{1}$$

where $S\left(w,D,C\right)$ is a score of word $w$ with respect to document $D$ and word cluster $C$ defined as:

$$S\left(w,D,C\right)=c\left(w,D\right)^{\alpha}\cdot s\left(w,C\right)^{\beta} \tag{2}$$

where $c\left(w,D\right)$ is the number of occurrences of word $w$ in document $D$, and $s\left(w,C\right)$ equals the similarity measure between vector representing the word $w$ and centroid of cluster $C$ if it contains $w$, otherwise 0. Parameters $\alpha$ and $\beta$ control the influence of the word count and similarity measure, respectively. The intuition behind (1) is that the score should be greater if the document has more words closer to the cluster centroid, as these words are assumed to be more related to the topic represented by the cluster. A similar intuition was used in [4], but there each cluster was reduced to a set of representative words closest to cluster centroid, only to compute the document-cluster score as the sum of frequencies of these words in the document. We found our definition to be more robust, as it eliminates the unfavorable case of a document not having any of its words in any of representative sets.

Another difference is that we make the *one-cluster-per-document* hypothesis, i.e., we assume that the most keywords of a document should be contained in a single cluster. We justify this hypothesis by the fact that the keywords are, by definition, related to the document's topic, and because the clusters are topically coherent, they should contain the majority of a document's keywords. Documents can be related to multiple topics, but usually these topics have a common ground. Parameter $k$, the number of clusters, also affects the size of clusters and hence the granularity of topics, so these related topics might be contained within the same cluster. Also, by restricting to a single cluster, we eliminate many words that are most probably not keywords because they are contained in other clusters and therefore are topically less related to the document in question. In an unfavorable case when the set of a document's keywords is equally distributed among two or more clusters, by choosing only one, recall is reduced but precision is preserved.

**Keyword Selection.** Keywords are selected from the intersection of words contained in the document and in the cluster the document has been tagged with. We tried two different selection methods. Both methods score each word $w$ with $S\left(w, D, C\right)$ defined by (2). The first method (keyword selection A) selects $n$ top-scored words, while the second method (keyword selection B) selects all words that have a score greater than $p \cdot S_{max}$, where $p \in [0, 1]$ and $S_{max}$ denotes the maximum score achieved by a word in a document.

### 3.2 Keyword-to-Keyphrase Expansion

Keywords extracted in the previous phase are now expanded into phrases that describe the document more precisely. Since each keyword might appear more than once in a document, multiple expansions are possible. Among the many phrases that represent a single concept, we prefer the most informative one. To this end we select to expand the first occurrence of a keyword in a document. The intuition behind this is that it is a common practice to reference a concept by its full phrase when it appears for the first time in a document, and later use an abbreviated form. The keyword-to-keyphrase expansion is done as follows:

1. Mark first occurrences of all selected keywords in the document;
2. If a noun or an adjective is marked, then unmarked adjacent nouns and adjectives that match the number, case, and gender of a marked word are also marked;
3. If an unmarked preposition is found between two marked words and if the second matches the case required by the preposition, then the preposition is also marked;
4. Repeat from step 2 until no further markings are made;
5. Every sequence of consecutive marked words is a keyphrase.

The above procedure expands the keywords into syntactically valid keyphrases in a conservative manner by appending only those words that agree with the keyword in number, case, and gender, as defined by the morphological lexicon [18]. By doing so, some keyphrases may be excluded or incomplete, but the expanded keyphrases are more likely to be syntactically and semantically valid.

## 4 Evaluation

Keyphrase evaluation is a complex and tedious task, mainly because of its high subjectivity. Also problematic is the number of extracted keyphrases and their specificity: does *"World Championships in Athletics"* make a better keyphrase than *"Championships in Athletics"*, or should we prefer to have both *"World Championships"* and *"Athletics"*? Some syntactic and morphological variation should also be taken into account, so that *"Championships in Athletics"* is treated as equivalent to *"Athletics Championships"* and *"Athletic Championship"*. Our evaluation methodology addresses most of these issues.

### 4.1 Evaluation Methodology

We base our evaluation on the comparison against keyphrases extracted by human experts in terms of the $F1$ measure, the harmonic mean of precision (P) and recall (R), i.e. $F1 = 2PR/(P + R)$ [16]. The evaluation set consist of 60 topically diverse newspaper articles provided by the Croatian News Agency. Average document length is 323 words, minimum 60, and maximum 1470 words. The documents were annotated independently by eight human experts, after which we computed the inter-annotator agreement in terms of the $F1$ measure. We then chose three out of eight annotators for which the inter-annotator agreement was the highest ($F1_{1,2} = 55.8\%, F1_{2,3} = 56.0\%, F1_{3,1}\% = 61.1$), and use the keyphrases extracted by these three annotators as the gold set. The remaining five annotators, compared against the gold set, establish a baseline of average human annotator performance.

The matching of extracted keyphrases against human extracted keyphrases follows the approach proposed by [19]. A pair of keyphrases is considered a match if one keyphrase is an inflectional morphological variant of the other, one keyphrase subsumes the other (partial match), or both. This accounts for differences in keyphrase specificity and morphological variation, but does not account for syntactic variation, which we found to be less prominent in the evaluation set. The justification for partial matches is that, even if the extracted keyphrase is more specific or more general, it still retains a part of the relevant information. Note that pairings between the two sets of keyphrases may be ambiguous; to resolve this we chose the pairings that optimize the total number of matches.

In order to account for the high subjectivity of the keyphrase extraction task, which is evident from the low inter-annotator scores, we compute the $F1$ measure asymmetrically as follows. We computed the precision with respect to the union of the results of the three gold standard annotators. Conversely, we computed the recall with respect to the intersection of the results of these three annotators. This makes the $F1$ measure tolerant to a false positive keyphrase assigned to a document by at least one of the three annotators, as well as a false negative keyphrase that is not assigned to a document by all three annotators.

### 4.2 Results and Discussion

Table 1 presents results of our keyphrase extraction method for chosen parameter values. We have tested both keyword selection methods: method A with parameter $n$ (the number of keywords to be selected) and method B with parameter $p$ (the ratio threshold). Other parameters are common to both methods: $N$ (the number of documents), $k$ (the number of clusters), and $\alpha$, $\beta$ (the weights used for tagging documents with clusters). The performance is measured in terms of precision ($P$), recall ($R$), the $F1$ measure, as described in the previous section.

Our method achieves a maximum value of $F1 = 44.5\%$, which is still lower than the worst human annotator, but comparable to supervised method using Naïve Bayes classifier [1] and outperforming the tf-idf keyphrase extractor [15].

The best results for both keyword selection methods are achieved when $k$ is set to a value close to the number of documents. For keyword selection A, increasing $n$ decreases precision and increases recall. For keyword selection B, decreasing $p$ has the same effect. This is expected because by doing so we increase the number of extracted keyphrases.

**Table 1.** Keyphrase extraction performance

| | $N$ | $k$ | $\alpha$ | $\beta$ | $n$ | $p$ | $P\,(\%)$ | $R\,(\%)$ | $F1\,(\%)$ |
|---|---|---|---|---|---|---|---|---|---|
| *Keyword selection A* | | | | | | | | | |
| * | 60 | 1 | 0.0 | 1.0 | 10 | – | 14.3 | 21.5 | 17.2 |
| | 60 | 1 | 1.0 | 1.0 | 10 | – | 26.4 | 43.8 | 33.0 |
| | 60 | 10 | 1.0 | 1.0 | 10 | – | 31.5 | 43.1 | 36.3 |
| | 60 | 50 | 1.0 | 1.0 | 4 | – | 53.1 | 37.7 | **44.1** |
| | 60 | 50 | 1.0 | 1.0 | 5 | – | 46.5 | 39.7 | 42.8 |
| | 60 | 50 | 1.0 | 1.0 | 14 | – | 31.0 | 53.6 | 39.3 |
| * | 60 | 60 | 0.0 | 1.0 | 7 | – | 33.3 | 33.1 | 33.2 |
| | 60 | 60 | 1.0 | 1.0 | 4 | – | 51.7 | 36.7 | 42.9 |
| | 60 | 60 | 1.0 | 1.0 | 5 | – | 47.2 | 38.9 | 43.1 |
| * | 60 | 60 | 1.0 | 0.0 | 7 | – | 37.3 | 39.8 | 38.4 |
| | 60 | 80 | 1.0 | 1.0 | 8 | – | 37.9 | 44.8 | 41.3 |
| | 100 | 120 | 1.0 | 1.0 | 10 | – | 35.3 | 48.0 | 40.6 |
| | 140 | 160 | 1.0 | 1.0 | 10 | – | 34.5 | 47.9 | 40.1 |
| | 200 | 200 | 1.0 | 1.0 | 10 | – | 34.1 | 46.2 | 39.2 |
| *Keyword selection B* | | | | | | | | | |
| | 60 | 1 | 1.0 | 1.0 | – | 0.6 | 32.5 | 19.2 | 24.2 |
| | 60 | 10 | 1.0 | 1.0 | – | 0.6 | 41.2 | 26.9 | 32.2 |
| | 60 | 60 | 1.0 | 1.0 | – | 0.4 | 33.3 | 45.7 | 38.5 |
| | 60 | 60 | 1.0 | 1.0 | – | 0.5 | 39.9 | 41.5 | 40.7 |
| | 60 | 60 | 1.0 | 1.0 | – | 0.6 | 53.3 | 38.2 | **44.5** |
| | 60 | 60 | 1.0 | 1.0 | – | 0.7 | 57.9 | 29.8 | 39.3 |
| | 60 | 60 | 1.0 | 1.0 | – | 0.8 | 58.9 | 28.5 | 38.4 |
| | 60 | 60 | 1.0 | 1.0 | – | 0.9 | 63.9 | 24.9 | 35.6 |
| | 100 | 120 | 1.0 | 1.0 | – | 0.6 | 50.8 | 35.1 | 41.5 |
| | 140 | 160 | 1.0 | 1.0 | – | 0.6 | 51.4 | 33.6 | 40.6 |
| | 200 | 200 | 1.0 | 1.0 | – | 0.6 | 49.0 | 32.6 | 39.1 |
| *Other approaches* | | | | | | | | | |
| | Naïve Bayes classifier [1] | | | | | | 39.5 | 52.3 | 45.0 |
| | Tf-idf keyphrase extractor [15] | | | | | | 29.5 | 64.6 | 40.5 |
| *Human annotators* | | | | | | | | | |
| | Average human annotator | | | | | | 68.2 | 63.2 | 65.1 |
| | Worst human annotator | | | | | | 64.2 | 48.5 | 55.2 |

When setting either $\alpha$ or $\beta$ in (1) to 0 (rows marked with an asterisk), i.e., ignoring the influence of word count and similarity measure, respectively, the performance decreases. A similar decrease is observed with different values of other parameters. This justifies the idea of combining topical relatedness with word count for boosting the performance of keyphrase extraction. Increasing the values of $N$ and $k$, while keeping their ratio nearly constant, does not impair the performance, making the proposed method applicable to larger corpora.

### 4.3 Evaluating Keyword-to-Keyphrase Expansion

A separate evaluation has been made for the keyword-to-keyphrase expansion. Two criteria have been evaluated: syntactic validity and keyphrase coverage. A phrase is syntactically valid if it represents a syntactic constituent. E.g.,*"archaeological museum"* is syntactically valid, whereas *"museum in"* is not. Our expansion procedure yields 89.44% syntactically valid phrases on the evaluation set. The keyphrase coverage tests how well the expanded keyphrases cover the keyphrases selected by the human annotators. Suppose the selected keyphrase is *"archaeological museum in Zagreb"*, whereas the expanded keyphrase is *"building of archaeological museum"*. In this case we have two true-positives (*"archaeological museum"*), two false-positives (*"building of"*), and two false-negatives (*"in Zagreb"*). Only those extracted phrases that share a common word with a human-selected keyphrase are compared. The keyphrase expansion achieves precision of 81.94%, recall of 79.73%, and $F1$ measure of 80.82%.

## 5 Conclusion

In this paper we presented an unsupervised approach to keyphrase extraction. The method builds clusters of topically related words, used to determine each document's keywords. The keywords are then expanded into phrases to describe the document more precisely. The experiments carried out on a Croatian dataset have shown that topical relatedness captured by the clusters does improve keyphrase extraction performance. The proposed method reaches the performance of $F1 = 44.5\%$, which is lower than the worst human annotator performance ($F1 = 55.2\%$), but comparable to supervised Naïve Bayes method.

There are several directions for further work. The topical coherence of clusters may be further improved by using other clustering methods or similarity measures. Another aspect that could be improved is the number of keyphrases to extract from a given document, as this number differs substantially from one annotator to another, and seems to depend on the length of the document. We would also like to evaluate our approach on other languages and domains.

**Acknowledgments.** This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under the Grant 036-1300646-1986. The authors thank the anonymous reviewers for their useful comments.

## References

1. Ahel, R., Dalbelo Bašić, B., Šnajder, J.: Automatic keyphrase extraction from Croatian newspaper articles. In: The Future of Information Sciences, Digital Resources and Knowledge Sharing, pp. 207–218 (2009)
2. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, Philadelphia, pp. 1027–1035 (2007)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of COLT 1998, pp. 92–100. ACM, New York (1998)

4. Delip, R., Deepak, P., Deepak, K.: Corpus based unsupervised labeling of documents. In: FLAIRS Conference, pp. 321–326 (2002)
5. Ercan, G., Cicekli, I.: Using lexical chains for keyword extraction. Information Processing and Management 43(6), 1705–1714 (2007)
6. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: Proc. of IJCAI 1999, pp. 668–673. Morgan Kaufmann Publishers Inc., San Francisco (1999)
7. Gulla, J.A., Borch, H.O., Ingvaldsen, J.E.: Unsupervised keyphrase extraction for search ontologies. In: Kop, C., Fliedl, G., Mayr, H.C., Métais, E. (eds.) NLDB 2006. LNCS, vol. 3999, pp. 25–36. Springer, Heidelberg (2006)
8. Hasan, K.S., Ng, V.: Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In: Coling 2010: Posters, Beijing, pp. 365–373 (2010)
9. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
10. Li, D., Li, S., Li, W., Wang, W., Qu, W.: A semi-supervised key phrase extraction approach: learning from title phrases through a document semantic network. In: Proc. of the ACL 2010, ACLShort 2010, pp. 296–300. ACL (2010)
11. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Proc. of NAACL 2009, pp. 620–628. ACL (2009)
12. Liu, Z., Li, P., Zheng, Y., Sun, M.: Clustering to find exemplar terms for keyphrase extraction. In: Proc. of EMNLP 2009, pp. 257–266. ACL, Singapore (2009)
13. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
14. McCallum, A., Nigam, K.: A comparison of event models for Naïve Bayes text classification. In: AAAI 1998 Workshop on Learning for Text Categorization, pp. 41–48. AAAI Press, Menlo Park (1998)
15. Mijić, J., Dalbelo Bašić, B., Šnajder, J.: Robust keyphrase extraction for a large-scale Croatian news production system. In: Proc. of FASSBL 2010, Dubrovnik, pp. 59–66 (2010)
16. van Rijsbergen, C.J.: Informaton Retrieval. Butterworths, London (1979)
17. Turney, P.D.: Learning to extract keyphrases from text. Tech. rep., NRC-IIT (2002)
18. Šnajder, J., Dalbelo Bašić, B., Tadić, M.: Automatic acquisition of inflectional lexica for morphological normalisation. Information Processing and Management 44(5), 1720–1731 (2008)
19. Zesch, T., Gurevych, I.: Approximate matching for evaluating keyphrase extraction. In: Proc. of RANLP 2009, pp. 484–489 (2009)
20. Zhang, K., Xu, H., Tang, J., Li, J.: Keyword extraction using support vector machine. In: Yu, J.X., Kitsuregawa, M., Leong, H.-V. (eds.) WAIM 2006. LNCS, vol. 4016, pp. 85–96. Springer, Heidelberg (2006)

# Voice Assessment of Speakers with Laryngeal Cancer by Glottal Excitation Modeling Based on a 2-Mass Model

Tobias Bocklet, Elmar Nöth, and Georg Stemmer

Lehrstuhl für Informatik 5 (Mustererkennung)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstr. 3, 91058 Erlangen, GERMANY
tobias.bocklet@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de

**Abstract.** The paper investigates the automatic evaluation of voice-related criteria of speakers with laryngeal cancer using a parametric two-mass model of the glottis. In contrast to previous approaches based on automatic speech recognition, the proposed method allows for a distinct evaluation of voice parameters alone since the underlying feature extraction technologies are based on a modeling of the whole vocal tract. This work focuses on the separation of vocal folds and vocal tract by LPC, where the vocal folds are represented by a parametric two-mass model which characterizes the excitation signal. The model parameters are optimized by a data-driven optimization procedure in order to fit the synthetic excitation signal to the LPC residue and the estimated pitch. We found first evidence that the computed parameters are meaningful in form of Pearson correlations between excitation signal parameters and different perceptual voice evaluation criteria in the range of $r \approx |0.7|$.

**Keywords:** Glottal excitation, voice modeling, perceptual evaluation.

## 1 Introduction

Voices of speakers with partial laryngectomy are often more hoarse, more harsh and more aspirated than normal speakers [11,7]. This can be explained by the anatomic alterations due to the cancer and/or the following treatment which may lead to a restricted movement of the vocal folds or to an inaccurate closure of the vocal folds [3,5,9]. In clinical routine, the quality of a person's voice is perceptually evaluated with respect to different rating criteria. Modeling the vocal folds by an adequate physical model may allow an automatic evaluation of distinct voice parameters.

This work focuses on the automatic evaluation of voice-related criteria on the basis of connected speech (read texts). In order to allow a (distinct) analysis of voices, an approach that is based on the source-filter model of the speech generation process is employed for voice evaluations. Voiced speech sounds are generated by the excitation signal, i.e., the source signal of the glottis. This signal is filtered by the vocal tract, where different frequencies are amplified or softened. In order to allow a meaningful evaluation of a person's voice, the influence of the vocal tract has to be omitted. This is achieved by assuming a linear filter between glottis and vocal tract. Linear prediction is

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 348–355, 2011.

applied to obtain the vocal tract configuration for each time frame. As an approximation of the excitation signal, the residue of the Linear Predictive Coding (LPC), an inverse filtering of the speech signal with the LPC filter is calculated in an data-driven optimization procedure. The model parameters are now optimized to match the synthetic excitation signal as close as possible to the LPC residue and the estimated pitch. The final parameters are then analyzed with respect to different voice evaluation parameters.

Automatic assessment of voice and speech criteria has already been investigated in previous works using different automatic speech processing techniques. [8], for instance, shows high correlations of different articulation and voice criteria between perceptual ratings and an automatic speech recognition (ASR) when a standard text is spoken. The usage of ASR systems for intelligibility assessment can be easily motivated: If the intelligibility of a speaker is low, the word recognition rate is low. A possible disadvantage of approaches based on ASR is that they ”=in contrast to the method presented in this paper”= account for the complete speech signal in form of spectral features. These features contain information of both the excitation signal of the vocal folds and the formant structure of the vocal tract/acoustic tube. This is not always intended, especially when it comes to the evaluation of distinct voice-related aspects.

The outline of this work is as follows: We first describe the used data in Chapter 2. The basics of the glottal excitation system are given in Chapter 3, the results are discussed in Chapter 4. The work is concluded by a summary and a short outlook in Chapter 5.

## 2 Dataset

Audio data were recorded from 85 patients (75 men, 10 women) suffering from cancer in different regions of the larynx. 65 of them had already undergone surgery with partial laryngectomy. They have been recorded 2.4 months after surgery on average. 20 speakers were still awaiting surgery. The average age of all speakers was 60.7 ±9.7 years. The youngest and the oldest person were 34 and 83 years old, respectively. Fig. 1 shows a patient before and after treatment of a T1 tumor of the right vocal fold. Before treatment the vocal fold oscillation of the right vocal fold is strongly limited. After surgery and radiotherapy the tumor is eliminated and the vocal fold oscillation has improved.

Each person read the text “Der Nordwind und die Sonne”, a phonetically balanced text with 108 words (71 disjunctive) which is used in German speaking countries in speech therapy. The English version is known as “The North Wind and the Sun” [4]. The speech data were sampled with 16 kHz and an amplitude resolution of 16 bit.

In order to obtain references for the automatic evaluation, five experienced phoniatricians and speech scientists evaluated each speaker regarding different voice (and speech) criteria. Voice quality, penetration, tone and intelligibility were rated regarding a 5-point scale with the labels very high, high, moderate, low, and none. Each raters decision for each patient was converted to an integer number between 1 and 5. Additionally roughness (R), breathiness (B) and hoarseness (H) were rated regarding the RBH-scale [13]. Evaluations are quantized from 0 (not present) to 3 (intense).

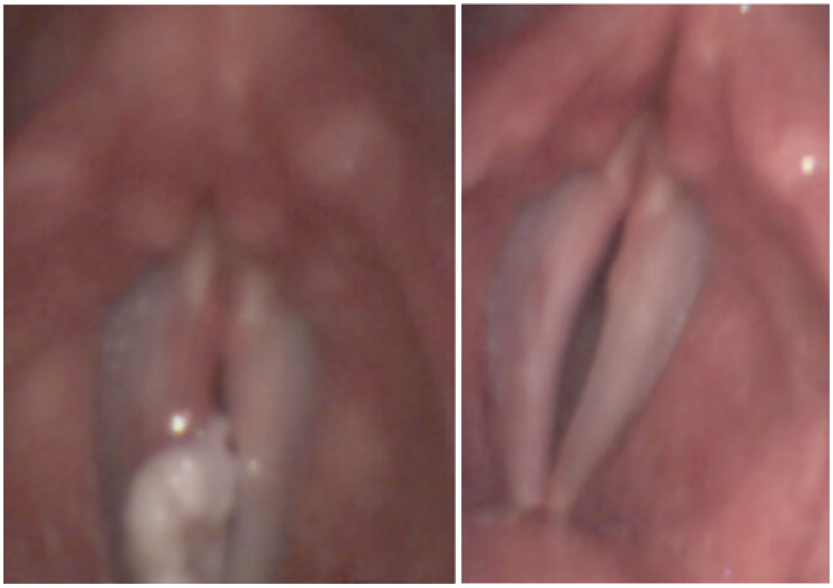


**Fig. 1.** Example of a person with a T1 tumor on the right vocal fold (left picture). After surgery and radiotherapy the tumor is eliminated and the vocal fold oscillation has improved (right picture). Oscillation capability has not recovered completely.

## 3 Glottal Excitation

### 3.1 Two Mass Model

The approach estimates the parameters of a physical glottis model from data of speakers with laryngeal cancer. The goal is to find pathology-related changes in the model parameters that reflect the voice quality and other voice related evaluation criteria. Therefore, the used glottis model should ideally have physically meaningful parameters, in contrast to just describing the shape of the excitation signal. The model should be flexible enough to adequately represent pathology-related changes of the voice quality.

Considering these requirements we employed the two-mass vocal fold model introduced by Stevens [12] and illustrated in Fig. 2. The model consists of two pairs of masses, larger ones ($M_1$) representing the inferior part of the vocal folds, and small ones ($M_2$) representing the superior part of the vocal folds. The model is symmetrical, there is no differentiation between the masses of the left and right side. The mechanism depends on the fact, that the inferior and superior part of the vocal folds do not move together as a rigid body. There is a certain degree of freedom to move relatively to each other [2]. This freedom is modeled by a coupling compliance by springs. Each mass moves on a spring that is connected with the latter wall. The masses are connected among themselves by an additional spring. The compliances of the springs are described by the parameters $C_1$, $C_2$ and $C_c$ (for the spring that connects $M_1$ with $M_2$). Note that parameters for the masses and compliances are given as *mass per unit length* and

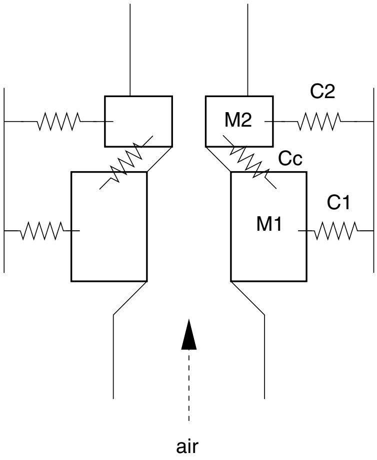


**Fig. 2.** Two-mass vocal fold model by Stevens [12]

*compliance per unit length*, i.e., they may change when the vocal folds are stretched. Air flows from bottom to top through the glottis when both $M_1$ and $M_2$ have a positive displacement, as shown in Fig. 2.

The excitation function of the two-mass vocal fold model by Stevens is obtained in three steps. First, the displacements $x_1(t)$ and $x_2(t)$ of the inferior and superior part of the vocal folds over time $t$ are computed. The width of the glottal opening $d(t)$ is defined to be $\min(x_1(t), x_2(t))$. Second, from the width of the opening, the airflow $U_g(t)$ through the glottis is determined. In the third step, taking the derivative of $U_g(t)$ results in the excitation function.

The whole process of the excitation function computation is described in Chapter 2 of [12]. However, some details cannot be found in the book. In [1] a detailed derivation of all model formulas is given. The initial and fixed values for all parameters are taken from [12] and summarized in [1].

### 3.2 Model Optimization

Our hypothesis is that glottis model parameters contain information about the degree of pathology of speakers with laryngeal cancer. To test this hypothesis, we find the optimal model parameters that fit the speech data and observe how they change with varying pathology.

Figure 3 depicts a block diagram of the optimization loop. A set of initial parameters ($M_1$, $M_2$, $C_1$, $C_c$, $x_0$, $d_1$, $\phi$, $l$) is the input of the glottis excitation model. The model generates an excitation signal for a 10 ms speech frame. At the same time, the LPC residue of the original speech signal is calculated and the log spectrum transform is applied to both of these excitation signals. The similarity of the generated excitation signal is compared to the original signal using two Euclidean distances. The distance between the log spectrum of the two signals is compared in a first step. In a second step, the distance between the generated and the original pitch for the frame are compared.

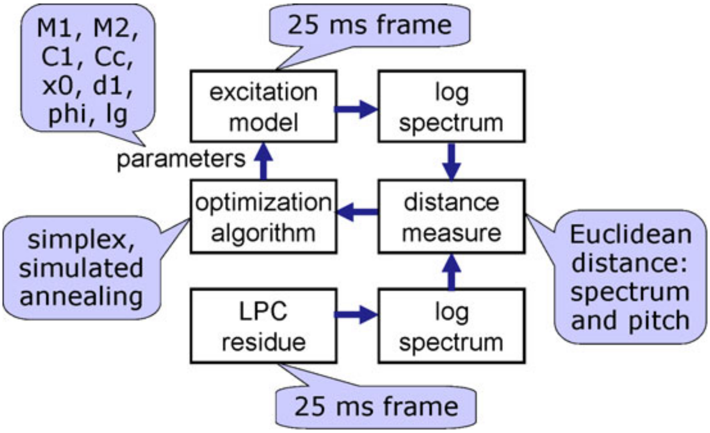


**Fig. 3.** Optimization of the parameters of the glottal excitation model

The combined distance measure is passed to the optimization algorithm, which modifies the parameter set, passing the new parameter set to the excitation model. Thus, an optimization loop is formed, modifying the parameters, generating a new candidate excitation signal, and testing it against the original signal. The simplex algorithm [10] and simulated annealing [6] are used for optimization.

The optimization is formulated as:

$$\begin{aligned}\hat{\theta} &= \underset{\theta}{\operatorname{argmin}}[D(s_m(\theta), s_{org})] \qquad (1)\\ \theta &= \{M_1, M_2, C_1, C_c, x_0, d_1, \phi, l\}\end{aligned}$$

where $D(s_m(\theta), s_{\mathrm{org}})$ is the combined distance between the model excitation signal $s_m$ and the original excitation signal $s_{\mathrm{org}}$.

The combined distance measure combines distances between both the respective log spectra and the respective pitches $p_m, p_{\mathrm{org}}$, and is defined as:

$$D(s_m(\theta), s_{\mathrm{org}}) = D(\mathrm{logspec}(s_m(\theta)), \mathrm{logspec}(s_{\mathrm{org}})) + \lambda \cdot D(p_m, p_{\mathrm{org}}) \qquad (2)$$

where $D(\cdot,\cdot)$ is the Euclidean distance between two vectors and the constant $\lambda$ scales the influence of the pitch distance. Note that the optimization is only performed on voiced speech segments.

Table 1 contains a description of all parameters of the glottal excitation model. The exact derivation of the formulas and parameters is omitted here. For description see [12] and [1].

The excitation parameters (Table 1) are calculated every 10 ms. We assumed that the standard deviations of the parameters, calculated per speaker over the whole text, are meaningful features for voice evaluations. In preliminary experiments we confirmed this. Pearson correlation coefficients are used as agreement measure.

**Table 1.** Description of the parameters of the glottal excitation model

| param | description |
|---|---|
| $M_1$ | mass of inferior part of the vocal fold |
| $M_2$ | mass if superior part of the vocal fold |
| $C_1$ | compliance of spring between M1 and lateral wall |
| $C_c$ | compliance of spring between $M_1$ and $M_2$ |
| $d_1$ | average vertical length of the lower portion of the vocal fold |
| $x_0$ | resting position of $M_1$ in the absence of any force |
| $\phi$ | skewness factor; representation of the constriction of the vocal tract |
| $l$ | length of glottis (assuming rectangular shape) |
| $D$ | optimization distance measure (see Eq. 2) |

## 4 Results and Discussion

In Table 2 the Pearson correlation coefficients among the different evaluation criteria is given. Pearson coefficients of $r > 0.9$ are measured between voice quality and tone, between tone and intelligibility, and between intelligibility and voice quality. Voice quality is highly connected to penetration, breathiness and hoarseness. The same statement holds also for tone and intelligibility. Among the RBH-scale breathiness and hoarseness and roughness and hoarseness correlate with $r > 0.8$. Roughness achieves only moderate ($r < 0.65$) correlations to voice quality, penetration, tone and intelligibility.

**Table 2.** Pearson correlation among the different perceptual evaluation criteria voice quality (quality), voice penetration (penetr), tone, intelligibility (intell), roughness (R), breathiness (B), hoarseness (H)

| | penetr | tone | intell | R | B | H |
|---|---|---|---|---|---|---|
| quality | 0.87 | 0.93 | 0.90 | 0.63 | 0.82 | 0.84 |
| penetr | | 0.84 | 0.86 | 0.45 | 0.73 | 0.70 |
| tone | | | 0.93 | 0.64 | 0.84 | 0.85 |
| intell | | | | 0.59 | 0.83 | 0.80 |
| R | | | | | 0.56 | 0.84 |
| B | | | | | | 0.81 |

The standard deviation of the excitation parameters are compared with the mean values of the perceptual evaluation criteria. The results are summarized in Table 3. We achieved moderate to good Pearson coefficients. Note that all of the correlation coefficients in Table 3 are negative. That means for example speaker with a high voice quality have a high standard deviation in masses $M_1$ and $M_2$. Note that the 9 excitation parameters are calculated every 10 ms. The variation between these 10 ms segments, i.e., phonemes, is high for speakers with a high voice quality. The changes of the parameters are lower between the 10 ms segments, when the speakers have a lower voice quality.

*Voice quality* correlates with $r = -0.69$ and $r = -0.67$ to the two masses $M_1$ and $M_2$. The Pearson coefficient between voice quality and the compliance of the spring between $M_1$ and $M_2$ ($C_c$) is $r = -0.71$. These three parameters achieve correlation coefficient in the same order for the criteria *tone* and *intelligibility*. This result is not really surprising, since these criteria correlate highly among each other (see Table 3).

The excitation parameter $\phi$ that represents the constriction of the vocal tract achieves a Pearson coefficient of $r = -0.69$ with the criterion *voice penetration*. People with a good *voice penetration*, have a high standard deviation of $\phi$, the constriction of the vocal tract changes a lot. This can be explained by a strong variation of the air flow. Note, that $\phi$ reaches such a high correlation only for the criterion penetration, for all other criteria it seems not to be an adequate feature.

The excitation parameters $M_1$, $M_2$ and $C_c$ achieve the highest correlation coefficients for the criteria of the RBH-scale. The differences in correlation coefficients of these three excitation parameters are not significant for these three criteria, nevertheless the mass $M_1$ achieves slightly better results. Breathiness achieved $r = -0.70$ and hoarseness achieved $r = -0.65$. Roughness showed only moderate correlations of $r = -0.41$. The length $l$ of the glottis shows moderate correlations $r \approx 0.6$ for most criteria. The parameters $D$, $d_1$, $x_0$ achieved only moderate correlations.

**Table 3.** Pearson correlation results between the perceptual evaluation criteria and the standard deviation of the parameters of the excitation system. The highest Pearson coefficient for each evaluation criterion is marked bold.

| param | quality | penetr | tone | intell | R | B | H |
|---|---|---|---|---|---|---|---|
| $M_1$ | -0.69 | -0.62 | -0.71 | -0.63 | **-0.41** | **-0.70** | **-0.65** |
| $M_2$ | -0.67 | -0.60 | -0.69 | -0.61 | -0.39 | **-0.70** | -0.63 |
| $C_1$ | -0.50 | -0.39 | -0.54 | -0.42 | -0.30 | -0.59 | -0.48 |
| $C_c$ | **-0.71** | -0.66 | **-0.72** | **-0.65** | -0.40 | -0.68 | **-0.65** |
| $\phi$ | -0.54 | **-0.69** | -0.51 | -0.51 | -0.23 | -0.44 | -0.42 |
| $l$ | -0.61 | -0.54 | -0.61 | -0.57 | -0.34 | -0.55 | -0.59 |
| $D$ | -0.53 | -0.44 | -0.54 | -0.47 | -0.27 | -0.54 | -0.49 |
| $d_1$ | -0.45 | -0.38 | -0.48 | -0.44 | -0.22 | -0.45 | -0.41 |
| $x_0$ | -0.24 | -0.23 | -0.27 | -0.27 | -0.07 | -0.20 | -0.22 |

## 5 Summary

In this work we applied a newly-developed glottal excitation system to the task of voice evaluations of speakers with laryngeal cancer. The system adapts different glottal parameters in a data-driven optimization loop to speech frames of 10 ms. We showed correlations between different parameters of the excitation system and speech evaluation criteria. The two masses and the compliance of the spring between these two masses showed good correlations to the parameters voice quality, penetration, tone, intelligibility breathiness and hoarseness. The parameter $\phi$ that represents the constriction of the vocal tract, showed the best correlation to the criterion penetration. In future work we plan to adapt more complex vocal fold models in order to achieve higher agreement between the model parameters and perceptual evaluations. Examples are the use of a non-symmetrical vocal fold model or pitch synchronous modeling.

**Acknowledgments.** The algorithm for the parameter estimation of the glottis model has been developed and implemented by Georg Stemmer and Andrew Cassidy at the Johns-Hopkins-Summerworkshop 2008. This study was partially funded by the German Research Foundation (DFG) under grant EY 15/18-2. We thank the speech therapists of our department for the perceptual evaluations.

## References

1. Beyerlein, P., Cassidy, A., Kholhatkar, V., Lasarcyk, E., Nöth, E., Potard, B., Shum, S., Song, Y.C., Spiegl, W., Stemmer, G., Xu, P.: Vocal aging explained by vocal tract modelling: 2008 JHU summer workshop final report. Tech. rep (2008)
2. Fant, G.: Acoustic Theory of Speech Production. Mouton, Netherlands (1960)
3. Fung, K., Lyden, T., Lee, J., Urba, S., Worden, F., Eisbruch, A., Tsien, C., Bradford, C., Chepeha, D., Hogikyan, N., Prince, M., Teknos, T., Wolf, G.: Voice and swallowing outcomes of an organ-preservation trial for advanced laryngeal cancer. Int. J. Radiat. Oncol. Biol. Phys. 63(5), 1395–1399 (2005)
4. Handbook of the International Phonetic Association. Cambridge University Press, Cambridge (1999)
5. Kim, C., Lim, Y., Kim, K., Kim, Y., Choi, H., Kim, K., Choi, E.: Vocal analysis after vertical partial laryngectomy. Yonsei. Med. J. 44(6), 1034–1039 (2003)
6. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by Simulated Annealing. Science 220(4598), 671–680 (1983)
7. Kosztya-Hojna, B., Rogowski, M., Pepiski, W., Rutkowski, R., Lazarczyk, B.: Voice analysis after the partial laryngectomy in patients with the larynx carcinoma. Folia histochemica et cytobiologica Polish Academy of Sciences Polish Histochemical and Cytochemical Society 39(Suppl 2), 136–138 (2001)
8. Maier, A., Haderlein, T., Eysholdt, U., Rosanowski, F., Batliner, A., Schuster, M., Nöth, E.: PEAKS - A system for the automatic evaluation of voice and speech disorders. Speech Communication 51(5), 425–437 (2009)
9. Makeieff, M., Barbotte, E., Giovanni, A., Guerrier, B.: Acoustic and aerodynamic measurement of speech production after supracricoid partial laryngectomy. Laryngoscope 115(3), 546–551 (2005)
10. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. The Computer Journal 7(4), 308–313 (1965)
11. Olthoff, A., Mrugalla, S., Laskawi, R., Fröhlich, M., Stürmer, I., Kruse, E., Ambrosch, P., Steiner, W.: Assessment of irregular voices after total and laser surgical partial laryngectomy. Arch. Otolaryngol Head Neck Surg. 129(9), 994–999 (2003)
12. Stevens, K.N.: Acoustic Phonetics. The MIT Press, Cambridge (1998)
13. Wendler, J., Rauhut, A., Krüger: Classification of voice qualities. Journal of Phonetics 14, 483–488 (1986)

# Web Text Data Mining for Building Large Scale Language Modelling Corpus

Jan Švec, Jan Hoidekr, Daniel Soutner, and Jan Vavruška

University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics
Univerzitní 8, 306 14 Plzeň, Czech Republic
{honzas,hoidekr,dsoutner,vavruska}@kky.zcu.cz

**Abstract.** The paper describes a system for collecting a large text corpus from Internet news servers. The architecture and text preprocessing algorithms are described. We also describe the used duplicity detection algorithm. The resulting corpus contains more than 1 billion tokens in more than 3 millions articles with assigned topics and duplicates identified. Corpus statistics like consistency and perplexity are presented.

**Index Terms:** language modelling, Internet, topic identification, duplicity detection.

## 1 Introduction

One of the key components of an automatic speech recognition (ASR) system is a language model. Training the language model requires a large number of training texts. The texts should be sampled from the target domain of the ASR system. It allows to recognize all domain-specific words and phrases and ensures that the out-of-vocabulary rate is low. For many domains it is possible to collect suitable large text corpora. It has been shown that increasing the size of training data leads to better performance. Therefore the data collection and preparation has the same priority as tuning the ASR system [1].

The data collection task is very common in many fields of research including automatic speech recognition [2] [3], speech synthesis [4] or natural language processing [5]. The very promising source of texts for language modelling task is the Internet. The huge number of Internet pages provides a very good basis for building large text corpora. Additionally the properties of metadata and hypertext links associated with web pages could be used during the corpus preprocessing. Research on using web pages for various natural language processing tasks is very extensive. For example Bulyko [3] uses Google queries to search for pages with a conversational style to recognize spontaneous speech in a telephone conversation and multiparty meetings. Another example for Czech is a work of Spoustová [5]. She used a web crawling method to obtain a large text corpora from Czech web pages.

In our work we decided not to collect random Internet pages but only to carefully select the Internet websites which provides a good language source for language modelling with respect to the target domain. The developed system and corpus could be used

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 356–363, 2011.

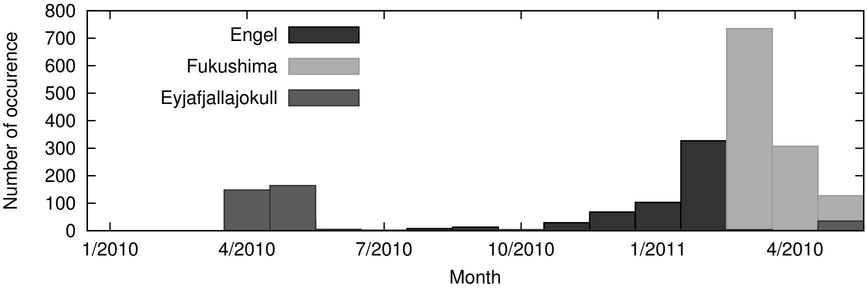


**Fig. 1.** The number of occurrences of some selected words by a month an article was published

for language modelling in an automatic subtitling system employing shadow speakers [6]. Therefore there is a need to lower the out-of-vocabulary rate which mainly influences the recognition accuracy and slows the subtitling process. The Internet news servers provide articles related to actual themes and the previously unseen words appear in the actual page texts very frequently. The large portion of the new words consists of names, geographic locations or names of causes. This is illustrated by the Fig. 1. The figure shows the number of occurences of three selected words in particular months starting year 2010. “Engel” is the name of a Czech union leader, “Eyjafjallajökull” is a well-known name of a volcano and “Fukushima” is a name of a city in Japan. It is obvious that the words related to actual causes can be words that have not been seen in the past and many of them have a non-standard pronunciation.

The presented system also contains a topic identification subsystem. It allows to select only texts related to some keyword or topic (group of keywords). The use of topics of stories for language model adaptation lowers the language model perplexity and a word error rate of the ASR system [2]. This paper describes a newly developed system which automatically collects text data from the Internet. In addition, it automatically runs text preprocessing task including a cleaning of web pages, a text tokenization, a text normalization and a vocabulary-based token substitution. The cleaned text is a subject of high-level text processing methods including a duplicity detection algorithm and a topic identification system. The text corpus created from the article database contains 1.13 billion tokens including punctuation marks (952 millions without punctuation). The database contains 3.1 millions articles and each article has a set of automatically assigned keywords. Additionally the duplicities are automatically detected and stored in the database.

## 2 System Architecture

Our system consists of an SQL database and a set of text processing algorithms which use the database as a data storage for the whole system. The algorithms are executed using the runtime environment which maintains a database connection and related tasks (transaction management, error recovery etc.). The separation of the data and the algorithms allows to easily develop new algorithms. The very important feature of our

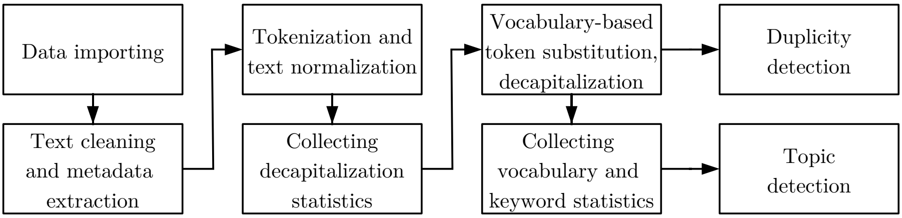

Data importing
Text cleaning and metadata extraction
Tokenization and text normalization
Collecting decapitalization statistics
Vocabulary-based token substitution, decapitalization
Collecting vocabulary and keyword statistics
Duplicity detection
Topic detection

**Fig. 2.** Schema of the architecture

architecture is a modularity. The database allows to easily introduce additional (often very complicated) relations between the database items. In other words, adding a new data structure to the system is very easy and does not influence any other parts of the system. The system architecture is depicted on Fig. 2. The entry-point of the system is the data importing module. It periodically checks predefined Internet sources for updates of related pages (Sec. 3). After getting a link to a newly published web page the cleaning and text preprocessing task starts (Sec. 4). During the web page cleaning a metadata extraction is performed and the corresponding database item is updated. The cleaned text is a subject for duplicity detection. Each run of the duplicity detection processes pages from a given time window and searches for pairs of items for which the duplicity relation holds (Sec. 5). Finally the topic identification is performed (Sec. 6).

## 3 Data Sources

Currently the system is able to process the web pages from three large Czech news websites: CeskeNoviny.cz (CNO), iDnes.cz (IDS) and Lidovky.cz (LID). Articles from CeskeNoviny.cz and Lidovky.cz contain mainly home and world news, business news, cultural and sport news. iDnes.cz also publishes local news, articles about hobby and living, cars, mobile phones and computers. Additionally we use the articles provided by the Anopress IT a.s. company (ANP) which contain news articles published in the printed news from Mladá Fronta and transcripts of the television news and discussion shows from the main Czech televisions and radio stations. The last source provides transcripts of the discussion show Questions of Vaclav Moravec (OVM) broadcast by Czech Television. The total number of articles and annual number of published articles are summarized in Tab. 2. The updates of the selected web sites are checked using the standard RSS format.

## 4 Text Preprocessing

### 4.1 Text Cleaning

The text cleaning algorithm in our system is a rule-based procedure which processes the input web page (an article mainly in the HTML format) and extracts the text of an article. Each of the data source is assigned a set of rules to extract the text and the metadata of the article. The metadata include the date when the article was published, keywords of the article, the author, the title and the subtitle etc. Embedded tables, images and text boxes are excluded from the further processing. In addition, the text is checked for invalid characters and character-based substitution is performed. The reduction of the character set simplifies the design of the subsequent processing algorithms.

### 4.2 Tokenization and Text Normalization

The text tokenization employs a rule-based method for dividing the text into a sequence of tokens. The special attention is paid to the tokenization of numbers (the decimal part of the number must not be split in a separate token) and electronic addresses. The punctuation marks are represented as standalone tokens. The tokenized text of an article is processed with a text normalization algorithm. It substitutes all occurrences of non-orthographical symbols (mainly numbers) with a corresponding full-length form, which could be processed by a phonetic transcription module. We used a normalization method described in [7].

### 4.3 Vocabulary-Based Token Substitution and Decapitalization

The tokens of a normalized text of an article are processed with a vocabulary-based substitution algorithm. The large vocabularies prepared by experts are used to normalize sequences of tokens. The substitution rules are of three types. The rules of the first type fix the common typos (eg. replacement of misspelled word “zda-li” to “zdali”). The rules of the second type replace sequences of tokens with a multiword (eg. a company name “Czech Coal” is replaced with “Czech_Coal”). The multiwords simplify the structure of a language model and lower its perplexity. The multiword rules correspond to names of known people, names of political parties and geographical names. The third type of rules unifies the written form of common terms (eg. a company name “EON” is unified with the correct form “E.ON”). A large number of terms has more than one rule because of inflection. Totally the human-prepared vocabularies contain 104k rules.

During the vocabulary-based substitution a statistical-based decapitalization is also performed. We use the term decapitalization for the process of substitution the capitalized words at the beginning of sentences with the corresponding lower-case variants for words other than the proper names or other word forms commonly written with the first letter capitalized. First a list of all capitalized words from the normalized text is extracted. For each of these words the ratio $d(w) = \frac{C'(w)}{C(w)}$ is computed from the normalized text. The $C'(w)$ is the number of sentences which start with the word $w$ and $C(w)$ is the number of occurrences of the word $w$ in the corpus. In other words, the statistic $d(w)$ expresses the ratio between the number of occurrences of the word $w$ after the punctuation mark or at the beginning of the paragraph and in the whole corpus. The words for which $d(w) \geq t_d$ (currently we are using $t_d = 0.9$) are decapitalized and the corresponding rules are added into the token substitution vocabulary.

## 5 Duplicity Detection

Due to the way the journalists work, the database contains a large number of duplicates. These duplicates are of two types. The first type is caused by republishing press material from official agencies or publishing the same article on two different web addresses (near duplicates). The second type of duplicates is caused by citing or merging older articles into a new one. The detection of duplicates is important because the language model created from a text including duplicates can prefer duplicated phrases and sentences instead of being a good model of the generic language. Our duplicity detection algorithm is based on the shingling method introduced by Broder [8] and allows to detect both types of duplicates. First an article is converted into a shingle set representation which is composed of a set of overlapping token bigrams. Then the metric rating the similarity of two shingle sets $A$ and $B$ is evaluated. The simplest similarity metric can be defined as a ratio of a number of shingles in both shingle sets to a number of shingles in a union of the two shingle sets:

$$S_1(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

The main disadvantage of this metric is a bad performance in cases where the length of $A$ is much different from the length of $B$ even though $A \subset B$. The solution is to introduce a *containment metric*:

$$S_2(A,B) = \frac{|A \cap B|}{|A|} \tag{2}$$

Unfortunately the $S_2$ metric is asymmetric ($S_2(A,B) \neq S_2(B,A)$). Therefore we use the symmetrized *maximum containment metric* defined by Malkin [9]:

$$S_3(A,B) = \max\{S_2(A,B), S_2(B,A)\} \tag{3}$$

This metric allows to compare shingle sets with a much different number of elements. The value of $S_3$ is from the interval $[0;1]$ where the value 0 means absolutely different shingle sets and the value 1 correspond with the cases where $A \subseteq B$ or $B \subseteq A$.

This definition of duplicity metric allows to define a *duplicity relation*. We say that an article (more precisely a shingle set) $A$ is a *duplicate* of an *original* article $B$ if $S_3(A,B) \geq t_s$ and $S_3(A,B) = S_2(A,B)$. Currently we are using $t_s = 0.5$. In other words, the shingle set $A$ is a duplicate of $B$ if there are half or more shingles from $A$ in the shingle set $B$ and the number of shingles in $A$ is lower than the number of shingles in $B$. For a very rare case $|A| = |B|$ we define that an newer published article is a duplicate and an older one is an original.

To construct the duplicity relation between $m$ articles we need $\frac{m(m-1)}{2}$ evaluations of Eq. 3. Also the storage space required to store a precomputed shingle set constructed from $n$-grams is $n$-times larger. Therefore we decided to use bigrams. We can also assume that the duplicates occur in a short time window so we detect duplicates only in a set of articles published in a window of two weeks. This detection is performed every day and each run of the detection processes up to 10k articles and takes approximately 7 minutes.

## 6 Topic Identification

One of the goals of our project was to develop a system which allows to train a topic-dependent language model. The topic identification task is possible because a large portion of articles in our database has assigned a set of keywords. Therefore we analyzed these keywords and we have found that the CNO source has the most reliable keywords. These articles and corresponding keywords were chosen as a train data for a naïve Bayes classifier and the articles from the other data sources were classified. From the set of all keywords assigned to the CNO data was selected a subset of keywords, which was manually clustered into a hierarchical tree structure. This structure allows to select articles with a given generic topic. For example the keywords *basketball*, *hockey* and *soccer* are groupped into a generic topic *sports*. The keyword structure has 32 generic topic (like *European union*, *transportation*, *sports*, *science*) and 474 leaf topics (keywords). The deepest path in the tree has a length of 4 nodes (for example *engineering* - *IT* - *computers* - *Apple*). According to our evaluation of topic identification experiments the classifier assigns exactly three keywords to each article. The complete topic identification algorithm and its evaluation is described in more detail in [10].

## 7 Corpus Statistics

We analysed the texts contained in the collected corpus. The number of articles, the number of tokens with and without punctuation marks (raw tokens) in dependence on the year of publication is shown in Tab. 1. The year *N/A* indicates that no publication date was available for the article. The main conclusion of this table is that our corpus contains over 1.1 billion tokens including the punctuation marks. The total number of articles, the number of articles published annually and the number of tokens divided by the source of an article is shown in Tab. 2. One can see that the ANP source has the largest volume of annually published articles. The IDS source has a smaller number of annually published articles but the total number of tokens is a one third of ANP. The articles from the source OVM are transcripts of weekly broadcast discussion television show, therefore it contains only 4 millions tokens but the data are very useful for modelling spontaneous discussion.

For the comparison of different data sources and for the evaluation of consistency of a particular data source we used a standard Spearman rank correlation coefficient of the distance of ranks of 500 most frequent words [5][11]. We can see that all data sources are very consistent (the value of the coefficient is higher then 0.9). Another observation is the high similarity between ANP and IDS sources. The cause is that some articles from IDS are published in the printed news MF Dnes which is included in the ANP source. The spontaneous speech contained in the OVM data is much different from each of the other sources (the coefficient is lower then 0.65).

Fig. 3 depicts the dependence of the size of the vocabulary on the number of tokens. The figure also shows the dependence of the size of a pruned vocabulary which contains only words occurring more than 5 times (resp. 10 times) on the number of tokens. The straight line represents a linear regression of the curve between 400M and 900M tokens. The extrapolation shows that by adding 1 million tokens of text to the corpus

**Table 1.** The number of articles and the number of tokens by a year of article publishing

| Year | Articles $\times 10^3$ | Tokens $\times 10^6$ | Raw Tkns $\times 10^6$ |
|---|---|---|---|
| N/A | 67 | 41.3 | 48.4 |
| 1998 | 26 | 6.2 | 7.1 |
| 1999 | 42 | 12.3 | 14.2 |
| 2000 | 172 | 52.5 | 63.3 |
| 2001 | 227 | 71.6 | 85.8 |
| 2002 | 239 | 73.9 | 88.8 |
| 2003 | 253 | 76.1 | 91.3 |
| 2004 | 282 | 80.2 | 96.0 |
| 2005 | 310 | 82.8 | 98.5 |
| 2006 | 324 | 89.2 | 106.1 |
| 2007 | 253 | 76.0 | 90.9 |
| 2008 | 261 | 82.7 | 98.7 |
| 2009 | 239 | 79.5 | 94.0 |
| 2010 | 304 | 96.6 | 114.6 |
| 2011 | 95 | 30.3 | 36.3 |
| Total | 3,101 | 952.0 | 1,134.1 |

**Table 2.** The number of articles and the number of tokens by a data source

| Source | # Articles $\times 10^3$ | Annually $\times 10^3$ | Tokens $\times 10^6$ |
|---|---|---|---|
| ANP | 2,354 | 184 | 647.1 |
| CNO | 125 | 48 | 44.3 |
| IDS | 537 | 48 | 225.0 |
| LID | 85 | 23 | 31.9 |
| OVM | 0.2 | 0.05 | 3.7 |

**Table 3.** The consistency and similarity of the data sources evaluated as Spearman rank correlation coefficients

| | ANP | CNO | IDS | LID | OVM |
|---|---|---|---|---|---|
| ANP | 0.97 | 0.91 | 0.98 | 0.97 | 0.63 |
| CNO | | 0.97 | 0.90 | 0.95 | 0.49 |
| IDS | | | 0.98 | 0.96 | 0.59 |
| LID | | | | 0.97 | 0.60 |
| OVM | | | | | 0.94 |

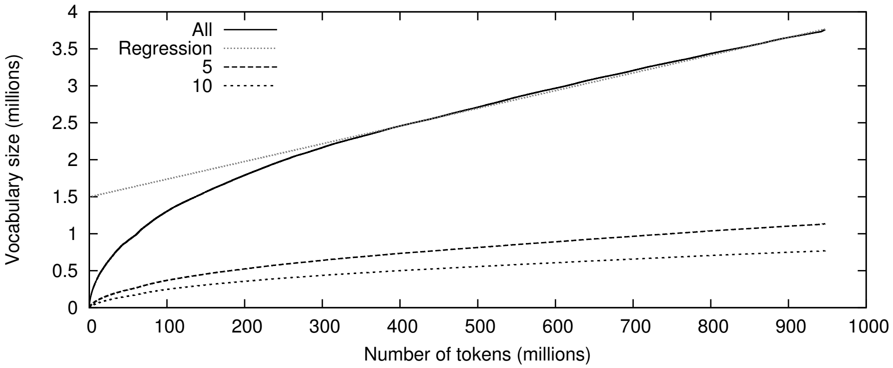


**Fig. 3.** The dependency of the vocabulary size on the number of tokens

the vocabulary grows by 2,400 new words. It fully corresponds with the Fig. 1 - a large number of previously unseen words in the news domain are used frequently only in a short time interval. The perplexity of a trigram language model trained from texts published in 2010 evaluated on texts published in January and February 2011 (21.7M tokens) is $PP_1 = 336.23$ with the OOV rate $0.66\%$ (absolutely 143k tokens). The perplexity of a model trained from texts published in 2010 evaluated on the same data is $PP_2 = 75.86$.

## 8 Conclusion and Future Work

We have presented a system which allows to build a large scale language modelling corpus. The used text preprocessing and duplicity detection methods were described. The statistics of the corpus with more than 1 billion tokens created from news articles were presented. They suggest that although the size of the collected corpus is relatively huge, there are still new words and topics arising and that it would be beneficial for keeping the language models up to date to continue with the collecting effort. In the future research we will focus on collecting much larger collections of data from many sources like discussion forums and movie subtitles using a statistical based text cleaning methods.

**Acknowledgments.** This work has been supported by the Ministry of Industry and Trade, project No. MPO FR-TI1/486, and by the grant of The University of West Bohemia, project No. SGS-2010-054. The access to the MetaCentrum computing facilities provided under the programme "Projects of Large Infrastructure for Research, Development, and Innovations" LM2010005 funded by the Ministry of Education, Youth, and Sports of the Czech Republic is appreciated.

## References

1. Müller, L., Psutka, J., Šmídl, L.: Design of speech recognition engine. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2000. LNCS (LNAI), vol. 1902, pp. 259–264. Springer, Heidelberg (2000)
2. Seymore, K., Rosenfeld, R.: Using story topics for language model adaptation. In: Proc. Eurospeech, vol. 97, pp. 1987–1990 (1997)
3. Bulyko, I., Ostendorf, M., Siu, M., Ng, T., Stolcke, A., Çetin, O.: Web resources for language modeling in conversational speech recognition. ACM Trans. Speech Lang. Process. 5 (2007)
4. 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)
5. Spoustová, D., Spousta, M., Pecina, P.: Building a Web Corpus of Czech. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010), Valletta, Malta (2010)
6. Trmal, J., Pražák, A., Loose, Z., Psutka, J.: Online TV Captioning of Czech Parliamentary Sessions. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 416–422. Springer, Heidelberg (2010)
7. Zelinka, J., Kanis, J., Müller, L.: Automatic transcription of numerals in inflectional languages. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 326–333. Springer, Heidelberg (2005)
8. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic clustering of the web. Computer Networks and ISDN Systems 29(8-13), 1157–1166 (1997)
9. Malkin, M., Venkatesan, R.: Comparison of texts streams in the presence of mild adversaries. In: Proceedings of the 2005 Australasian Workshop on Grid Computing and e-research, ACSW Frontiers 2005, vol. 44, pp. 179–186. Australian Computer Society, Inc. (2005)
10. Skorkovská, L., Ircing, P., Pražák, A., Lehečka, J.: Automatic topic identification for large scale language modeling data filtering. In: Habernal, I., Matoušek, V. (eds.) TDS 2011. LNCS(LNAI), vol. 6836, pp. 64–71. Springer, Heidelberg (2011)
11. Kilgarriff, A.: Comparing corpora. International journal of corpus linguistics 6(1), 97–133 (2001)

# Web-Based System for Automatic Reading of Technical Documents for Vision Impaired Students*

Jindřich Matoušek[1], Zdeněk Hanzlíček[1], Michal Campr[2], Zdeněk Krňoul[1], Pavel Campr[1], and Martin Grůber[1]

[1] University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics,
Univerzitní 8, 306 14 Plzeň, Czech Republic

[2] University of West Bohemia, Faculty of Applied Sciences,
Dept. of Computer Science and Engineering,
Univerzitní 8, 306 14 Plzeň, Czech Republic

**Abstract.** A web-based system for automatic reading of technical documents focused on vision-impaired primary-school students is presented in the paper. An overview of the system, both its backend (used by teachers to create and manage the documents) and frontend (used by students for viewing and reading the documents), is given. Text-to-speech synthesis utilised for the automatic reading and, especially, the automatic processing of mathematical and physical formulas are described as well.

**Keywords:** web-based system, automatic reading of technical documents, text-to-speech, reading of mathematical formulas, vision impaired.

## 1 Introduction

In this paper, a contribution to the integration of modern web with speech and language technologies is presented. More specifically, automatic reading of technical documents within the project ARET (Automatic Reading of Educational Texts for Vision Impaired Students) is introduced. The project aims at an innovation and enhancement of schooling of vision impaired (both purblind and blind) primary-school students and also at a facilitation of their self education. Technical documents include, but are not limited to, topics of Mathematics and Physics (ISCED 2 level).

Within the project a web-based system for automatic reading of technical documents was developed. Teachers use the system for a preparation, management and administration of educational texts. The texts are available to students online via system's frontend; they are read aloud by means of text-to-speech synthesis (TTS).

From the point of view of TTS, an automatic reading of technical documents and, especially, mathematical and physical formulas is very challenging, at least

* Project ARET CZ.1.07/1.2.00/08.0021 is co-funded by the European Social Fund and the State Budget of the Czech Republic. Support was also provided by the University of West Bohemia, project No. SGS-2010-054.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 364–371, 2011.

from two reasons. Firstly, text processing of formulas (i.e. decoding and transcription of their symbolic notation to a corresponding word form) is not trivial and requires a special treatment different from processing in a general-purpose TTS system. Secondly, most current TTS systems are based on a corpus-based approach to speech synthesis in that they are optimised on the corpora utilised during the system design. As mathematical and physical formulas are usually not included in these corpora, problems with synthetic speech quality (both intelligibility and naturalness) can arise when read automatically.

Speech- and language-processing technologies enable to develop assistive applications for people with various impairments or health disorders, see e.g. [1] for an example of a medicine application or [2] for an example of a system for deaf people. Some projects for reading technical documents or mathematical formulas for the vision impaired also exist. The problem of reading mathematics has been already solved, e.g. in the system AsTeR (Audio System for Technical Readings) [3] or in the system AudioMath developed at Porto University [4]. For the Czech language, the Lambda editor was created at Masaryk university (http://www.teiresias.muni.cz/czbraille8). Within the presented project ARET, a new system for reading mathematical formulas is being developed.

The paper is organised as follows. In Section 2, an overview of the developed system for automatic reading of technical documents is presented. The text-to-speech system used for reading the texts aloud is briefly described in Section 3. Special issues related to this specific utilisation of TTS technology are depicted in Section 4. Finally, conclusions are drawn in Section 5.

## 2 System Overview

The developed system is a web application, based on client-server architecture, running on *Apache* HTTP server with *MySQL* database system. The core of the system is based on *Symfony 1.4*, an open-source web application framework.

The system is divided into two separate sections: frontend and backend. Frontend serves as a public interface for viewing various technical documents (arranged as topics) and, at the same time, reading them. On the other hand, backend is an administrative interface, where the documents can be created and modified. A schematic view of the system can be seen in Figure 1.

### 2.1 Backend

System's backend is shown in Figure 2. Teachers have a direct access to the backend through a *WYSIWYG* text editor in which they create and modify the technical documents. Within the project ARET, the editor was enhanced with the ability to insert project-specific templates and formulas. The templates are used to clarify the meaning of a particular fragment of the document, for example the *Important* template is for highlighting a crucial information to which the students should pay more attention, or the *Example* template is used to display various examples. Different synthetic voices can be assigned to each template. Currently, five templates are supported: Definition, Important, Note, Example, and Solution.

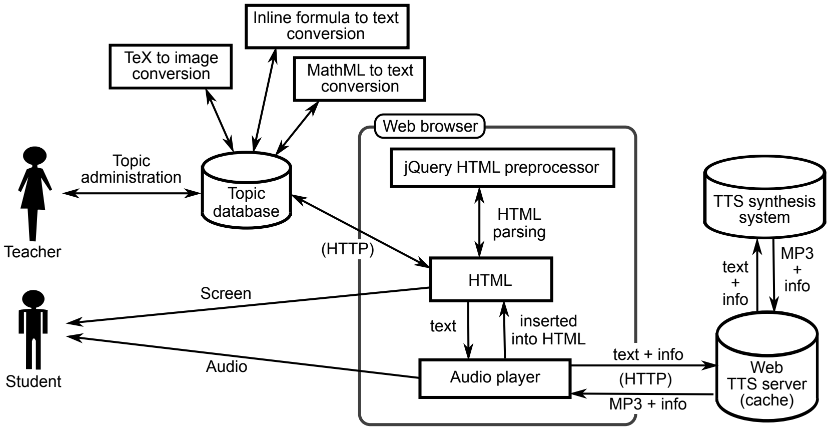
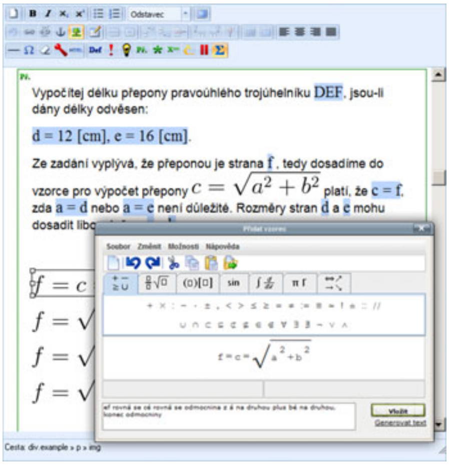


**Fig. 1.** System diagram



Vypočítej délku přepony pravoúhlého trojúhelníku DEF, jsou-li dány délky odvěsen:

d = 12 [cm], e = 16 [cm].

Ze zadání vyplývá, že přeponou je strana f, tedy dosadíme do vzorce pro výpočet přepony $c = \sqrt{a^2 + b^2}$ platí, že c = f, zda a = d nebo a = c není důležité. Rozměry stran d a c mohu dosadit libo

$f = c$

$f = \sqrt{}$

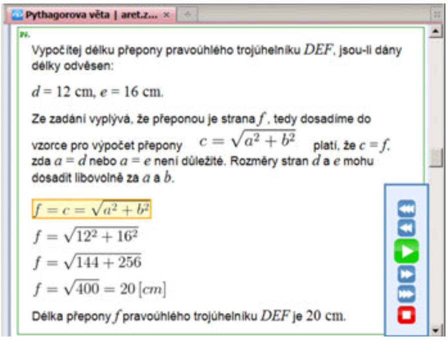
$f = \sqrt{}$

$f = \sqrt{}$

$f = c = \sqrt{a^2 + b^2}$

ef rovná se cé rovná se odmocnina z á na druhou plus bé na druhou, konec odmocniny

**Fig. 2.** Backend – text editor and formula editor



Vypočítej délku přepony pravoúhlého trojúhelníku $DEF$, jsou-li dány délky odvěsen:

$d = 12$ cm, $e = 16$ cm.

Ze zadání vyplývá, že přeponou je strana $f$, tedy dosadíme do vzorce pro výpočet přepony $c = \sqrt{a^2 + b^2}$ platí, že $c = f$, zda $a = d$ nebo $a = e$ není důležité. Rozměry stran $d$ a $e$ mohu dosadit libovolně za $a$ a $b$.

$$f = c = \sqrt{a^2 + b^2}$$

$$f = \sqrt{12^2 + 16^2}$$

$$f = \sqrt{144 + 256}$$

$$f = \sqrt{400} = 20\,[cm]$$

Délka přepony $f$ pravoúhlého trojúhelníku $DEF$ je 20 cm.

**Fig. 3.** Frontend – text which is currently being read is highlighted by the yellow colour; audio player interface is in the bottom right corner of the web page

There are two different ways for inserting mathematical formulas into the document. Simple formula with linear structure like $x + 1$ can be written as "inline formula". To write more complex formulas, *DragMath* editor which enables to store a formula in both *MathML* and TEX formats (with MathML being used to derive a word-level description of the formula for speech synthesis—see Section 4—and with TEX being used to generate an image of the formula for displaying in the system's frontend) was adopted.

### 2.2 Frontend

System's frontend is a public web interface where the topics are displayed and read aloud to students. Before displaying the web page, the HTML documents are automatically processed—optimised for TTS-based synthesising (the content texts, including the processed formulas, are extracted).

The texts are then sent to the *Web TTS server* which is responsible for the automatic reading. The text-to-speech conversion is made by the TTS system (described in Section 3) in which texts are synthesised as MP3 (or OGG) files. System's cache is also supported to avoid re-synthesising already synthesised texts. To play the audio files, JavaScript player using *Adobe Flash* or *HTML5 <audio> tag*, in which paragraph- and section-based navigation is supported, is employed.

## 3 Text-to-Speech

For the automatic reading of technical documents in system's frontend, Czech TTS system ARTIC [5] has been employed. ARTIC applies a corpus-based concatenative speech synthesis method. Based on a carefully designed speech corpus (a collection of a large number of utterances annotated on orthographic, phonetic and prosodic levels), statistical approach (with hidden Markov models, HMMs) was employed to perform an automatic phonetic segmentation of the source speech corpus into phones. Based on this segmentation, boundaries between diphones, the basic speech units used in the ARTIC system, were located. As a result, acoustic unit inventory (AUI), the source speech corpus indexed with diphones and prosodic structures, was built.

During run-time speech synthesis, phonetic and prosodic aspects of an input text are estimated first. Ideally, input text is a subject of a thorough analysis and processing. Due to a complexity of such a task, current text processing in the ARTIC system is somewhat simplified to four main steps: text normalisation of "non-standard" words (digits, abbreviations, acronyms, etc.) [6], detailed rule-based phonetic transcription, including pronunciation dictionary of "exceptional" words (mostly foreign words, names, physical units, etc.) [5], phones-to-diphones conversion, and prosodic description in terms of prosodic symbols (prosodic clauses, phrases, prosodemes, etc.) using prosodic phrase grammar [7]. Within the scope of the ARET project, text normalisation also includes the processing of mathematical and physical formulas described with the inline formulas or MathML codes. Such specially marked formulas could be then processed and converted to words (see Section 4).

Prosodic analysis includes punctuation-driven sentence clause detection, rule-based word stress detection and symbolic prosodic description [7]. Symbolic features based on a prosodic phrase grammar, like prosodic sentence, prosodic clause, prosodic phrase, prosodic word, and prosodeme, were used to describe prosodic characteristics and to express prosodic structure of to-be-synthesised texts.

The resulting speech is generated by a *unit-selection* algorithm [8]. Its principle is to smoothly concatenate (according to *join cost*) speech segments (diphones in our case), extracted from natural utterances using the automatically segmented boundaries, from large speech unit inventories according to phonetic and prosodic criteria (*target cost*) imposed by the synthesised utterance. As there are usually many instances of each speech segment, there is a need to select the optimal (with respect to both target and join costs) instances dynamically during synthesis run-time (using the unit-selection technique). To calculate the target cost, a prosodic structure of the to-be-synthesised utterance is estimated, and a comparison between prosodic symbolic features (plus some positional features, like position of a diphone in a prosodic word, and contextual factors like immediate left and right phone) in the utterance and in the unit inventory is carried out. Join cost is evaluated as a distance between spectral features and pitch around the concatenation point of two potentially neighbouring speech units. After selecting the optimal sequence of (diphone) speech segments, neither prosodic nor spectral modifications are made in the ARTIC system except for simple smoothing at concatenation points. To cope with high CPU power and memory cost typical for unit-selection systems, a computational optimisation was carried out as described in [9].

## 4 Automatic Reading of Formulas

Text processing is an important part of a TTS system. Generally, text processing in a TTS system depends on the type of texts that are likely to appear at the input of the system. In the ARET project, educational texts (currently the texts of Mathematics and Physics at ISCED 2 level—i.e. mathematical and physical formulas) are expected as an input of the TTS system.

Reading of mathematical formulas in the Czech language is a very complex task, especially if the problem is supposed to be solved generally, i.e. there is no limitation for the complexity of the equation structure. In fact, any final system will be naturally limited by the definition of expected mathematical operations, types of operands, etc. Nevertheless, the system should be simply extensible by additional definition of reading rules, e.g. for new operators.

Since Czech is an inflective language, the correct grammatical form of particular operands (numbers and variables) can be different in various mathematical contexts. For the inflection, methods described in [6] were employed.

As mentioned in Section 2, two different representation of mathematical formulas are employed in our system. Simple formulas with a linear structure can be written and stored as a simple text. For the creation of more complex mathematical expressions, the special editor DragMath is employed and their structure is represented by using an MathML format. In both cases, thanks to a special syntax or marking of the formulas in the HTML code, no detection of formulas is needed.

### 4.1 Reading of "Inline" Formulas Represented by a Plain Text

Formulas written by a text ("inline formulas") have a simple linear mathematical structure—it is usually a sequence of operators and operands, which can be read in the same order as it is written. Mathematical priority of particular operation has no significance for the reading. All operands in the formula have to be inflected into the corresponding grammatical form, which is determined by the previous operator (the first operand is in its primal form). We define a simple transcription rule for each operator, which contains

- transcription for a given operator
- grammatical form for the following operand (case, number and gender)

By the sequential application of those rules and operand inflection (described in [6]), the whole formula can be read.

In the current version, only some basic operators and operand types are supported in the text representation, e.g. addition, subtraction, multiplication, division, brackets, superscript (power), subscript, numbers, variables and physical units. For formulas with other operators or with a more complex structure, MathML representation has to be employed.

### 4.2 Reading of Formulas Represented by MathML

MathML (http://www.w3.org/TR/MathML3) is an XML application for describing mathematical notation. It is capable to represent mathematical formulas of almost any structure and complexity. Moreover, the standard can be easily extended with new operators or operand types. For purposes of our project, we defined a special operand type for physical units, an operator for applying operations on both sides of equation, etc.

The transcription of formulas represented by MathML can be divided into several steps:

- hierarchical decomposition of a MathML code
- selecting suitable transcription rules for particular operator
- applying the selected rules (operator transcription and the corresponding inflection of related operands)

For each mathematical operation, several transcription rules can be defined. They differ by their activation conditions, i.e. in various mathematical contexts, for various values or types of operands, different transcription rules can be selected. For most operators, one basic rule and several additional rules for exceptional cases are defined.

Each transcription rule contains a text template for the resulting expression together with the corresponding grammatical form for each operand (case, number, gender, cardinal or ordinal form, etc.). Moreover, a type of the resulting expression is also defined because this expression could be an operator in the higher level of the hierarchical structure representing the whole formula.

An illustrative (incomplete) example of transcription rules for two operators—power and fraction (in YAML notation):

```
POWER:
- condition: { operand_2_type: [ number, variable ] }
  operands:
  - { type: cardinal, case: 1, number: S, gender: F }
  - { type: ordinal, case: 4, number: S, gender: F }
  template: "{operand_1} na {operand_2}"
  expr_type: expression

FRACTION:
- condition: { any_operand_type: [ fraction, fraction_expression ] }
  operands:
  - { type: cardinal, case: 1, number: S, gender: F }
  - { type: cardinal, case: 1, number: S, gender: F }
  expression: "složený zlomek, nad hlavní zlomkovou čarou je
            {operand_1}, pod hlavní zlomkovou čarou je {operand_2}"
  expr_type: fraction_expression

- condition: { any_operand_type: [ expression, function ] }
  operands:
  - { type: cardinal, case: 1, number: S, gender: F }
  - { type: cardinal, case: 1, number: S, gender: F }
  expression: "zlomek, v čitateli je {operand_1}, ve jmenovateli je {operand_2}"
  expr_type: fraction_expression

- condition: { operand_1_type: [ number, variable ],
               operand_2_type: [ number, variable ] }
  operands:
  - { type: cardinal, case: 1, number: S, gender: feminine }
  - { type: cardinal, case: 7, number: S, gender: feminine }
  expression: "{operand_1} lomeno {operand_2}"
  expr_type: fraction
```

An example of MathML representation and transcription of a given formula follows. Aforementioned rules are applied in this example.

| **Formula** | **MathML representation** | **Transcription** |
|---|---|---|
| $\frac{x^n}{y^3}$ | `<math xmlns="http://www.w3.org/1998/Math/MathML">`<br>` <mfrac>`<br>`  <mrow><msup>`<br>`   <mrow><mi>x</mi></mrow><mrow><mi>n</mi></mrow>`<br>`  </msup></mrow>`<br>`  <mrow><msup>`<br>`   <mrow><mi>y</mi></mrow><mrow><mi>3</mi></mrow>`<br>`  </msup></mrow>`<br>` </mfrac>`<br>`</math>` | zlomek, v čitateli je iks na entou, ve jmenovateli je ypsilon na třetí |

Particular rules for each operator are ordered and their condition evaluated from most special to most general. The last rule has usually an empty condition; thus, it is applied always when none from preceding rules is selected. It is quite easy to define a new set of transcription rules or extend an existing one with rules for new mathematical operations or with additional rules for some rare linguistic exceptions.

The conversion of formulas to text is shown in Figure 1 in blocks *"Inline formula to text conversion"* and *"MathML to text conversion"*.

## 5 Conclusion and Future Work

Automatic reading of technical documents within the project ARET, a contribution to the integration of modern web with speech and language technologies, was presented in the paper. Since ARET focuses on Mathematics and Physics, automatic processing of mathematical and physical formulas was dealt with. Nevertheless, the system framework has been designed to be general and flexible enough to cover also other kinds of technical documents, including more advanced topics like tertiary level of mathematics, etc. Although the ARET project is still being worked on, the first technical documents are already available on http://ucebnice.zcu.cz.

Future work will be focused on three main areas. First, the number of topics will be continuously increasing in order to cover the intended goal of the ARET project. Second, system functionality is also planned to be enhanced. For instance, other rules for reading formulas will be added. We also plan to personalise the system for each user by allowing him/her to change the layout of web pages with topics (e.g. colours of fonts, templates, etc.). Finally, we will also try to make the developed system more compatible with other tools and systems that vision impaired use. For instance, we will unify keyboard shortcuts.

## References

1. Hippmann, R., Dostalová, T., Zvarová, J., Nagy, M., Seydlová, M., Hanzlíček, P., Kříž, P., Šmídl, L., Trmal, J.: Voice-supported Electronic Health Record for Temporomandibular Joint Disorders. Methods Inf. Med. 49(2), 168–172 (2010)
2. Krňoul, Z., Kanis, J., Železný, M., Müller, L.: Czech Text-to-Sign Speech Synthesizer. In: Popescu-Belis, A., Renals, S., Bourlard, H. (eds.) MLMI 2007. LNCS, vol. 4892, pp. 180–191. Springer, Heidelberg (2008)
3. Raman, T.V.: Audio System for Technical Readings. Ph.D. Thesis, Cornell University, New York (1994)
4. Ferreira, H., Freitas, D.: Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 678–685. Springer, Heidelberg (2004)
5. 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)
6. Zelinka, J., Kanis, J., Müller, L.: Automatic Transcription of Numerals in Inflectional Languages. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 326–333. Springer, Heidelberg (2005)
7. 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)
8. Tihelka, D., Matoušek, J.: Unit Selection and Its Relation to Symbolic Prosody: a New Approach. In: Proceedings of Interspeech, Pittsburgh, USA, pp. 2042–2045 (2006)
9. Tihelka, D., Kala, J., Matoušek, J.: Enhancements of Viterbi Search for Fast Unit Selection Synthesis. In: Proceedings of Interspeech, Makuhari, Japan, pp. 174–177 (2010)

# Zanzibar OpenIVR: An Open-Source Framework for Development of Spoken Dialog Systems

Dmytro Prylipko[1], Dirk Schnelle-Walka[2], Spencer Lord[3], and Andreas Wendemuth[1]

[1] Chair of Cognitive Systems, Otto-von-Guericke University Magdeburg
Magdeburg, Germany
{dmytro.prylipko, andreas.wendemuth}@ovgu.de
[2] Telecooperation Lab, Darmstadt University of Technology
Darmstadt, Germany
dirk@tk.informatik.tu-darmstadt.de
[3] Spokentech, Inc., San Francisco, CA
spencer.lord@gmail.com

**Abstract.** The maturity of standards and the availability of open source components for all levels of the MRCP stack provide us with new opportunities for the development of spoken dialog technology. In this paper a standard-based and modular architecture for interactive voice response (IVR) systems is presented together with its implementation – Zanzibar OpenIVR. The architecture, described in terms of components and standards, is compared to other existing frameworks. The usage of our framework is discussed regarding different aspects of spoken dialog technology such as speech recognition and synthesis, integration of the components, dialog management, natural language understanding. It is designed to work over VoIP as well as with usual telephony communication channels, thus provides an ability for web based access. Zanzibar OpenIVR is able to serve as a starting point for building dialog systems and research in voice-enabled technologies.

**Keywords:** IVR, VoiceXML, VoIP, Spoken dialog system.

## 1 Introduction

Since Bolt's *Put that there* [1] speech is considered to be among the most important means of post desktop interaction. Considerable progress in the development of voice based interfaces has been achieved during the recent years, bringing us to natural access to information.

Following the W3C voice browser activity, state-of-the-art spoken dialog systems provide functions such as call handling, speech recognition and synthesis, retrieval of data and dialog management together with language understanding. Looking ahead, modern systems should provide access via VoIP as well as existing communication lines (GSM and PSTN). IP telephony is more flexible than regular PSTN lines and provides us with ample opportunities for call routing and management. Moreover, it allows integration with services available over the Internet, e.g. with click-to-call technology, thus aiming at what is generally called the *voice-enabled web*.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 372–379, 2011.

Recently, several proprietary frameworks have been developed for building spoken dialog systems. Whereas prices are quite reasonable, we adhere to an opinion that open source solutions not only save money, but also encourage developers and scientists to study and improve technology. Open standards for a particular field as well as the corresponding software, thereby offer a playground for modifications and full control during the development stage.

A recent study [2] showed that there are also serious security issues using third party software. The conjunction with the findings of Hammonds et al. from Forrester Research [3] who claim that the average bug fixing time for enterprise developers is 6.9 days while 36% of the open source developers need under 8 hours to fix a bug since it was discovered, shows the potential of open source vs. proprietary software.

Although a significant number of open source application frameworks have been designed in the recent years, these solutions are often in-house tools, tied to particular products or custom APIs, which leads to huge efforts for learning and using such systems.

That is why special attention should be given to resort to open standards and pluggable components in order to make the whole framework transparent, flexible and reusable. Additionally, these standards should be suitable to be used in research projects by enabling to rely on functional building blocks while keeping the opportunity to modify the default behavior, to go beyond the state of the art.

In this paper we describe such a modular open source framework for spoken dialog systems that meets these special requirements within research and development. Thereby, we address the need for a common test-bed. We also provide a small overview of existing systems.

## 2 The Architecture of Zanzibar OpenIVR

Zanzibar OpenIVR[1] uses the JVoiceXML interpreter and contains a MRCPv2 Server with the Sphinx4 speech recognizer and FreeTTS speech synthesizer. It integrates with a VoIP PBX (Private Branch Exchange) (like Asterisk) using SIP and RTP VoIP standard. It can deploy and run VoiceXML documents as well as applications written in Java.

An overview of the Zanzibar's architecture is presented in Fig. 1. A more detailed description of component interactions is given in section 2.2.

### 2.1 Components

**Telephony Server.** The telephony subsystem functions as the gateway to the voice network. It provides access to various telephony protocols and networks. It does routing and call handling for incoming calls, provides flexible dialplan management. It must be a robust and scalable server and should have a rich API and extension framework that provides the ability to add modules that can inetgrate with other components. It should be based on open source solution for the advantages over proprietary solutions mentioned above.

[1] http://sourceforge.net/projects/openivr/

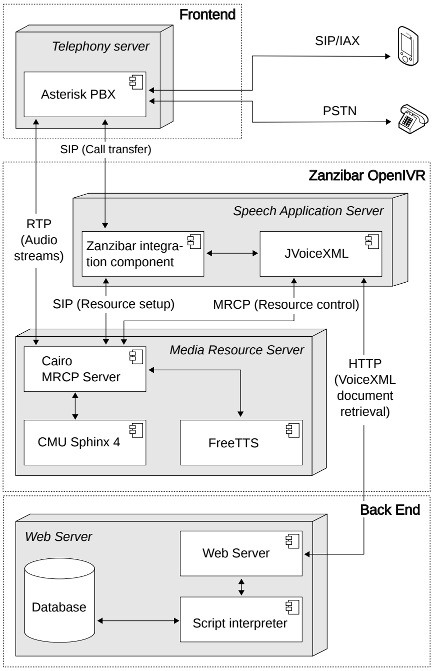


**Fig. 1.** Architecture of a spoken dialog system built on Zanzibar OpenIVR

Our proposed solution employs Asterisk[2] as a telephony interface because of its popularity, continuous development and reliable support.

**Speech Application Server.** After the call has been received by the telephony server, the processing of the dialog starts. The small survey of established frameworks in section 3 shows that VoiceXML is suitable to fulfill our request for a standardized dialog control component. As a domain specific language, VoiceXML allows for task independent, modular and reusable dialog development. The major advantages of using VoiceXML are:

- It is an open W3C standard independent of any existing implementation (it has the status of W3C recommendations since 2000). It is supported by a set of powerful tools for dialog development;
- VoiceXML applications are less expensive in development compared to traditional IVRs [4].
- It provides mechanisms for dynamic management of the spoken dialog.

JVoiceXML[3] is an open source VoiceXML browser written entirely in the Java programming language, supporting the VoiceXML 2.1 standard.

[2] http://www.asterisk.org/

[3] http://www.jvoicexml.org/

Besides the support of Java APIs such as JSAPI and JTAPI, custom speech engines can easily be integrated into this platform. Examples are a text based platform and an MRCPv2 platform that are shipped with the voice browser.

**Media Resource Server.** The Media Resource Control Protocol (MRCP) is a communication protocol designed to provide a mechanism to control processing resources on the network by a client. These resources are usually speech recognizers (ASR engines) and speech synthesizers (TTS engines). MRCP allows the implementation of distributed interactive voice systems, for instance VoiceXML interpreters. In such a system, the voice browser acts as an MRCP client, while the MRCP server provides an access to ASR and TTS engines. It has additional functionality such as load balancing, clustering and failure handling in order to meet the scalability requirements. The key benefit in the separation of the VoiceXML interpreter and the media resource server is the independence of components and their ability to easily be replaced.

Zanzibar OpenIVR uses Cairo[4] as a media resource server. It implements a subset of the MRCP version 2 specification and integrates with CMU Sphinx 4 and FreeTTS which are described below.

**Speech Recognition Engine.** For speech recognition the CMU Sphinx 4 [5] is employed. The recognizer is a state-of-the-art continuous-speech, speaker-independent systems based on hidden Markov models (HMMs) and N-gram statistical language models.

CMU Sphinx 4 provides numerous capabilities: generalized pluggable frontend and language model architectures, rich capabilities for language modeling, generalized acoustic model architecture, utilities for post-processing recognition results (obtaining confidence scores, generating lattices), speaker adaptation mechanisms. CMU Sphinx is distributed under an academic BSD-style license. The code and binaries are free for commercial and non-commercial use.

**Text-to-Speech Engine.** FreeTTS[5] is an open source speech synthesis system based upon CMU's Flite, which is by-turn derived from the Festival and the FestVox project, from Carnegie Mellon University. FreeTTS is released under BSD license.

FreeTTS supports a number of voices (including MBROLA voices) and is also able to import vocies from FestVox. It implements the speech synthesis part of JSAPI 1.0.

## 2.2 Integration Issues

Figure 1 shows how all the components integrate together in the Zanzibar OpenIVR to form a coherent system. Zanzibar consists of three servers: the teltphony server, the speech application server and the media resource server.

The PBX server acts as a gateway between the users and Zanzibar. When the call is to be routed to the IVR, the transfer utilizes the SIP protocol by the `Dial` command

[4] `http://www.speechforge.org/projects/cairo/`

[5] `http://freetts.sourceforge.net`

from the Asterisk dialplan to connect the telephony server with the speech application server. The use of the SIP protocol enables the use of any other standard PBX.

The speech application server interacts with the telephony platform, establishes, maintains and tears down sessions, runs the application for the user and gathers speech resources as needed from the MRCPv2 server.

VoiceXML functionality is enabled by the use of JVoiceXML running as an embedded component that is hooked via the Zanzibar integration component. Speech recognition and synthesis are not provided by JVoiceXML itself, but by the external systems, so-called *implementation platforms*. JVoiceXML has a framework to add different implementation platforms. The primary mechanism to do this is through a system out- and user input Service Provider Interface (SPI). Such an implementation has been done in the MRCP4J client library, which is used to integrate JVoiceXML with its implementation platform - Cairo MRCP server.

As one can see from the Fig. 1, the audio streams go directly from the media resource server to the telephony server bypassing the speech application server. The audio is transferred to the PBX and back via RTP protocol. CairoRTP library over Java Media Framework (JMF) is used for that.

### 2.3 Dialog Management and Natural Language Understanding

An analysis in [6] names three dialog strategies that can be used in the development of voice user interfaces: user initiative, system initiative and mixed initiative.

VoiceXML claims to support all three, although not all are explicit goals. The most common dialog strategy that is used in current telephony applications is system initiative. User initiative dialogs are rarely used, although they are possible to develop. VoiceXML is designed around the concept of *form items* that are divided into *input items* where the user can enter some data and *control items* containing executable code. Both input and control form items are designed to support the development of system initiative dialogs.

In a mixed initiative dialogs the user can take the initiative. If the system can not fulfill the request from the user because the request is missing some data, the system can take the initiative to ask for the missing data. In this case the system essentially falls back to form filling mode. This comes closer to the vision of a human-like conversation. In this case the concepts of VoiceXML are still valid. The user's input is still restricted to match certain grammars. Hence, information overloading is not possible. Also, there is no real support for intention recognition to determine the user's goal for calling. It remains restricted to the current dialog context. Summing up, the way VoiceXML treats mixed-initiative is not what is really mixed-initiative but slot-filling.

With Zanzibar we are also limited to the concepts of VoiceXML. However, many applications based on this technology have been successfully developed and deployed. Efficient dialogs, coming close to a natural interaction, are only possible with carefully designed dialog scripts. Many guidelines tell how to achieve that, like the one given in [7] exist. There are also first attempts to transfer the idea of design patterns [6] to the design of voice user interfaces.

## 3 Related Work

In the recent past, several solutions for spoken dialog have been presented. The Thai voice application gateway [8] is similar to the framework proposed in this paper. Unfortunately, this branch has not been ported back to the origin JVoiceXML source code and the entire system is not available anymore because of technical reasons.

Olympus [9] is a framework for spoken dialog system created at Carnegie Mellon University (CMU). At the high level it consists of a series of components connected in a pipeline architecture. Several spoken dialog systems have been successfully developed and deployed using Olympus framework (Let's Go!, LARRI, TeamTalk etc.).

The Jaspis open framework [10] provides a radical change in the way how spoken dialog systems are developed, moving away from frame-based application design. Hence, it establishes its own proprietary standard with a limited user group, while we rely on established standards to address a broader community.

Noteworthy solutions of commercial software are the Sympalog speech technology [11], InterpreXer VoiceXML IVR Server[6], OptimTalk[7] and framework built with Asterisk + VXI* VoiceXML browser[8] together with supported ASR and TTS engines. A framework presented in [12] also provides a platform for development of multipurpose dialog systems, but is built with some commercial software.

Only four of the frameworks mentioned above (Thai voice application gateway, Voxy over Asterisk, OptimTalk and VXI*) are based on a standardized programming interface for dialog control and modeling which is VoiceXML in all cases.

Whereas one can see that number of commercial as well as research solutions for spoken dialog systems is quite significant, we have to state that no full open-source stack based on VoiceXML 2.1 standard is available.

## 4 Evaluation and Discussion

The work reported in this paper results from an integration of existing open source components with open standards into the particular framework, namely Zanzibar OpenIVR. The cooperation with several research related institutes prove Zanzibar to be a good starting point for the development of voice based applications.

In Table 1 a comparison of several currently available frameworks is presented from the point of view of flexibility, openness and integration capabilities. Under integration with IP-PBX we mean the ability of the framework to deal with the telephony system via standard protocol like SIP or IAX. The system is considered to be modular if it allows to interchange components from different vendors, which can be achieved, for instance, using standards. For instance, Jaspis and Olympus are open, modular and transparent, however based on in-house infrastructure which hampers integration.

The key feature of the described framework is its flexible nature: It relies on standards developed for the spoken dialog technology and does not depend on a particular implementation therefore. In its current configuration (Asterisk, JVoiceXML, Cairo,

[6] http://www.phonologies.com/interprexer.php

[7] http://www.optimsys.cz/technology/introduction.php

[8] http://www.i6net.com/products/vxi/

**Table 1.** Comparison of the frameworks for spoken dialog systems

| Framework | Integration with IP-PBX | Modular | VoiceXML based | Open source |
|---|---|---|---|---|
| InterpreXer VXML Server | + | + | + | – |
| Jaspis | + | +/– | – | + |
| Olympus | – | +/– | – | + |
| OptimTalk | + | + | + | – |
| Sympalog | + | + | – | – |
| VXI* VoiceXML Browser | + | + | + | – |
| Zanzibar OpenIVR | + | + | + | + |

Sphinx, FreeTTS) it presents one possible combination of components. Other ones can be created as needed.

All of the related frameworks evaluated in this paper can be used to evaluate things like dialog strategies, acoustic and language models within that particular framework. But the standards-based, modular nature of the Zanzibar design provides an additional advantage over the other related frameworks. It can also be used for comparing various components, e.g. speech recognition engines or VoiceXML interpreters on the same task.

It exposes standards and provides a testbed for their development by common effort. Commercial software also provides such a possibility, but often in a limited amount: a support for a non-standard component almost always requires a corresponding update by request.

## 5 Conclusion and Further Work

In this paper we described Zanzibar OpenIVR – an open source framework for the development of telephone-based voice-enabled applications. It can be useful in research and development of communication standards, components of conversational systems, testing dialog flows and acoustic models etc. However some shortcomings remain, mostly related to technical imperfections. In its future development we will try to make it faster, more reliable and easier in installation and configuration. These goals can be accomplished with further improvements like pooling of VoiceXML sessions to decrease the response time or less complicated installation procedure with auto-registration in Asterisk.

We hope that the overview of frameworks for conversational systems given in this paper will make the landscape of proposed solutions less daunting, and help researchers choose the most appropriate platform for a given task.

## References

1. Bolt, R.A.: Put-that-there: Voice and gesture at the graphics interface. SIGGRAPH Comput. Graph. 14, 262–270 (1980)
2. Veracode: State of software security report volume 2. Research report, Veracode (2010)
3. Hammond, J.S., Gerush, M., Sileikis, J.: Open source software goes mainstream. Research document, Forrester Research (2009)

4. Jackson, E.: Speaking up for cost savings in the call center: Vxml takes on the dinosaur of legacy ivr (2003), http://www.thefreelibrary.com/Speaking+up+for+cost+savings+in+the+call+center:+VXML+takes+on+the...-a0107216561 (last accessed 08/20/2010)
5. Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., Wolf, P., Woelfel, J.: Sphinx-4: a flexible open source framework for speech recognition. Technical report, Mountain View, CA, USA (2004)
6. Schnelle, D.: Context Aware Voice User Interfaces for Workflow Support. PhD thesis, TU Darmstadt (2007)
7. Cohen, M.H., Giangola, J.P., Balogh, J.: Voice User Interface Design. Addison-Wesley, Boston (2004)
8. Kaitrungrit, D., Dailey, M.N.: Thai voice application gateway. In: Proceedings of ECTI-CON 2008, pp. 101–104. IEEE, Los Alamitos (2008)
9. Bohus, D., Raux, A., Harris, T.K., Eskenazi, M., Rudnicky, A.I.: Olympus: an open-source framework for conversational spoken language interface research. In: NAACL-HLT 2007: Proceedings of the Workshop on Bridging the Gap, pp. 32–39. Association for Computational Linguistics, Morristown (2007)
10. Turunen, M., Hakulinen, J.: Jaspis – a framework for multilingual adaptive speech applications. In: Proceedings of the 6th International Conference on Spoken Language Processing, Beijing (2000)
11. Nöth, E., Horndasch, A., Gallwitz, F., Haas, J.: Experiences with Commercial Telephone-based Dialogue Systems (Erfahrungen mit kommerziellen Telefon-Sprachdialogsystemen). It - Information Technology 46(6), 315–321 (2004)
12. Nuno, J.N., Neto, J.P., Mamede, N.J., Cassaca, R., Oliveira, L.C.: The Development Of A Multi-Purpose Spoken Dialogue System. In: Proceedings of EUROSPEECH (2003)

# Building Support Tools for Russian-Language Information Extraction

Mian Du, Peter von Etter, Mikhail Kopotev,
Mikhail Novikov, Natalia Tarbeeva, and Roman Yangarber

Department of Computer Science,
University of Helsinki, Finland
{mian.du,peter.etter,mikhail.kopotev,mikhail.novikov,
natalia.tarbeeva,roman.yangarber}@cs.helsinki.fi

**Abstract.** There is currently a paucity of publicly available NLP tools to support analysis of Russian-language text. This especially concerns higher-level applications, such as Information Extraction. We present work on tools for information extraction from text in Russian in the domain of on-line news. On the lower level we employ the AOT toolkit for natural language processing, which provides modules for morphological analysis and partial syntactic chunking. Since the outputs of both lower-level modules contain unresolved ambiguity, we synthesize the outputs and pass the result into a pre-existing English-language analysis pipeline. We describe how the information extraction system is adapted for multi-lingual support, including extensions to the ontologies and to the pattern matching mechanism. While this is work in progress, we present an end-to-end pipeline for event extraction from Russian-language news.

## 1 Introduction

We describe work in an on-going project, to adapt the PULS information extraction (IE) system, [3,2] to extend an existing English-language IE system, to extract structured content from Russian-language on-line news. While the IE system has been applied to many different news domains, in this paper we focus specifically on the *border-security* domain.

The English-language IE system contains modules for morphological, shallow–syntactic and semantic analysis. For Russian, the system requires analogous components. Building morphological and syntactic analyzers from scratch is infeasible in a short time-span, due to the immense complexity of the language; therefore we tried to use existing tools for this purpose. However, at present, there is a dearth of publicly available tools for Russian-language natural language processing (NLP). This especially concerns higher-level applications, such as Information Extraction (IE), but is true as well of tools for lower-level analysis. For example, it is reported that the University of Sheffield GATE system, [4], which supports multi-lingual IE, has been adapted to Russian as part of the MUSE-3 project, but there is little information available on its functionality.

After a thorough evaluation of the available linguistic resources for Russian, [1], we chose the AOT toolkit, (www.aot.ru), as the most promising of the existing freely-available tools. The situation with resources is somewhat better; for example, we

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 380–387, 2011.

incorporated a comprehensive geographic gazetteer available as part of the multi-lingual GeoNames database, (www.geonames.org). In this paper, we describe the current status of the project, the components integrated so far, and outline next steps for building a system for analysis of Russian text.

## 2 Background and Context of the Work

We now briefly describe a specific context in which the Russian-language analysis tools are being applied, which serves as a motivating case-study.

At present various government authorities acknowledge that a significant amount of information useful for monitoring situations relating to public safety is publicly available in the form of published material on the Internet. This has led to an interest in advanced tools that combine techniques from text mining, machine learning, statistical analysis and computational linguistics to help analysts and intelligence experts to manage the growing volume of information, to filter out the irrelevant material, and to extract valuable knowledge from on-line sources.

We collaborate with partners, including the European Agency for the Management of Operational Cooperation at the External Borders of the Member States of the European Union, to facilitate the process of extracting structured information on events related to border security from on-line news articles. A particular focus is placed on incidents of illegal migration, cross-border criminal activity, and crisis situations at the EU borders. The rationale for exploiting on-line media sources for this purpose is threefold. First, information on certain border security-related events might not be available from other sources, or it might be incomplete in those sources. Second, such information might be available from other sources (e.g., through dedicated networks), but there might be significant delays before the information passes through official channels. Third, open-source information on-line media, can be used for cross-verification against information obtained from other sources.

The need for strengthening the capabilities for tracking the security situation for illegal migration has been identified and acknowledged by the European Commission (EC)[1]. Specifically, the Commission Communication COM (2008) 68A proposes the creation of an Integrated European Border Surveillance System (EUROSUR), and suggests development and deployment of new tools for strategic information to be gathered from various sources (e.g., from open sources) in order to recognize patterns and analyze trends, supporting the analysis of migration routes and the prediction of risks.

The specifications of the EUROSUR network outlined above impose certain requirements on the tools for on-line news event extraction, in particular, they should: extract information in real or near-real time, extract as fine-grained event descriptions as possible, process news articles in many different languages, since a large fraction of relevant events are only reported in non-English, local news.

In this setting, special emphasis is placed on multi-lingual processing, since information about various geographic areas is typically published in different languages,

[1] *Examining the creation of EUROSUR*, `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2008:0068:FIN:EN:PDF`.

and for the system to be useful and to have sufficient coverage, application to multiple languages is an important requirement.

To address these needs, we undertook the work described in this paper.

## 3 Baseline English System

As a basis, we use the PULS IE system, described in, e.g., [12]. PULS is similar in design in many aspects to GATE, and it had been adapted to various domains; however, to date the support in PULS has mainly focused on English-language text [7,5]. PULS contains modules for lower-level (morphological and syntactic) as well as higher-level (semantic) analysis, and at the end of the pipeline produces filled *templates* extracted from an input corpus. An output template is a *structured* description of a real-world event, in the subject domain. For example in the border-security domain, a news article about smuggling of illegal materials between two countries should induce an event of type "*smuggling*", to which the system should attach the names of the locations/countries involved, the date of the incident, the perpetrators, the type and amount of goods smuggled, etc.

The link from syntactic to semantic analysis is provided by pattern matching: a pattern is a sequence of elements to be matched on input text. The elements can be stated in terms of syntactic, morphological, as well as semantic constraints. A matched pattern invokes an *action*, which can group matched elements into higher-level objects, and eventually into events; e.g., a noun phrase is composed of nouns with modifying adjectives and numerals—*four illegal immigrants*, etc.

**Ontologies and Concepts:** in defining IE patterns, it is common to group concepts into ontologies, to improve coverage. For example, the concept "contraband" would include the sub-classes "drug", "weapon" and "animal", each of which, in turn contains many sub-concepts; then a single pattern can be stated in terms of the high-level concept, to capture all kinds of contraband.

**Inference Rules:** Patterns match sequences of words in a sentence. A higher-level mechanism for detecting events is *inference rules*, similar to those employed in expert systems. The job of a pattern is to transform "syntactic" objects—words in the sentence—into "logical" or semantic objects. Inference rules operate strictly at the logical level, at a higher level of abstraction. For example, an inference rule may state that if a smuggling event is mentioned in the text, and there is a mention of known drug or weapon within one sentence from the mention of the event, we can (probably) assume that it refers to the smuggled items. We implemented an inference rule module, based on one described in [14].

## 4 AOT

The AOT project ("automated processing of text" in Russian) grew out of the DIALING project, [10], which was a commercial project on automatic translation, ended in 2001. Subsequently, components that were used for linguistic analysis were transformed into the AOT toolkit, [9], released under the open-source GNU LGPL license. AOT is a

**Table 1.** Output of Lemm, the AOT morphological analyzer, adapted with English labels (morphological ambiguity preserved)

| Byte | Surface | Lemma | POS | Morphological tags |
|---|---|---|---|---|
| 0 | На | на | Prep | — |
| 3 | берегу | беречь | Finverb | Impf Transv Act Pres 1p Sg |
| 3 | берегу | берег | Noun | Inan Masc Sg {Dat\|Loc} |
| 10 | пограничной | пограничный | Adj | Fem Sg Anim Inan {Gen\|Acc\|Inst\|Loc} |
| 22 | реки | река | Noun | Inan Fem {Sg Gen\|Pl Nom\|Pl Acc} |
| 27 | задержано | задержать | SParticip | Perf Transv Anim Inan Past Pass Sg Neut |
| 36 | двадцать | двадцать | Card | {Nom\|Acc} |
| 45 | семь | семь | Card | {Nom\|Acc} |
| 50 | нелегалов | нелегал | Noun | Anim Masc Pl {Gen\|Acc} |

**Table 2.** Output of Synan, the AOT shallow syntactic analyzer, adapted with English labels

| Relations | | | | | | |
|---|---|---|---|---|---|---|
| Type | Parent | | | Child | | |
| | ID | Surface | Lemma | ID | Surface | Lemma |
| Num-Noun | 7 | нелегалов | НЕЛЕГАЛ | 5 | двадцать семь | — |
| Adj-Noun | 3 | реки | РЕКА | 2 | пограничной | ПОГРАНИЧНЫЙ |
| Gen-Nom-Group | 1 | берегу | БЕРЕГ | 3 | реки | РЕКА |
| Prep-Group | 0 | На | НА | 1 | берегу | БЕРЕГ |

| Groups | |
|---|---|
| Type | Members |
| Cardinal-Ordinal-Group | двадцать(5) семь(6) |

collection of modules for natural language processing, including libraries for morphological, syntactic, and semantic analysis, language generation, tools for working with dictionaries, and GUIs for visualizing the analysis. In work described below, we use only the morphological and syntactic analyzers, called *Lemm* and *Synan*. (The module for semantic analysis appears to be unfinished, [11].) These analyzers needed to be adapted for our purpose, to correct certain inaccuracies and to output more information, as they were originally designed for different purposes. A major complicating factor resulting from the evolution of the project is incomplete documentation.

In the examples that follow, we use a simple input sentence На берегу пограничной реки задержано двадцать семь нелегалов , ("twenty seven illegal migrants have been detained on the bank of the borderline river").

**Lemm Morphological Analyzer:** The output of Lemm's morphological analysis is shown in table 1. The columns indicate the byte offset of the word, the surface form, the lemma (or base form), the part of speech, and the morphological tags. As appropriate for a pure morphological analyzer, Lemm does not attempt to resolve ambiguity, and passes it downstream. For example, the surface form берегу, derives from the lemma for the noun "(river) bank", but may also be an inflection of the verb "preserve." Finer ambiguity, on the level of morphological tags, is indicated by |, as when the case is ambiguous for a given lemma.

**Synan Syntactic Analyzer:** Synan attempts to generate a complete syntactic dependency parse tree; in general it produces a collection of tree fragments. Synan output for the sample sentence is shown in table 2. Synan identifies binary parent-child relations, and "groups"; a group is a sequence of words which function syntactically as an atom, and are not analyzed for dependency (e.g., "twenty seven"). In the process, it resolves morphological ambiguity.

# 5 System Integration

For building the Russian-language IE system, we integrated the following phases:

- Identify and pre-categorize Russian documents
- Linguistic analysis: morphology, syntax, semantics
- Filling event slots

The input documents are gathered by a dedicated Web crawler, [8], which harvests news articles in Russian matching a large list of keywords that are indicators of potential relevance for the target domain—in this paper, cross-border crime, such as drug smuggling or human trafficking.

## 5.1 AOT Wrapper

The document text is next processed by the AOT tools. Neither Lemm nor Synan alone extract sufficient syntactic information from the text for building patterns: Lemm, because it does not resolve ambiguity, and Synan, because it does not process all words, only those that participate in recognized relations/groups. Thus we wrapped these two modules into a single, combined analyzer. The purpose of the wrapper is to output a complete analysis of all the words in the sentence, with as much ambiguity removed as possible, and as many relations identified as possible. The wrapper goes through the following stages.

**Pre-processing of Lemm and Synan Output:** We parse the XML-like outputs of Lemm and Synan into structure shown above, in tables 1 and 2. The part-of-speech (POS) tags and morphological and syntactic tags are mapped into common English tags, (as shown), from the original AOT-specific encoding.

We then normalize the groups to look like other binary relations (by chaining the words in the group), and correct certain inconsistencies or inaccuracies that we have identified in AOT; for example, the analysis of patronymics in Russian names was not appropriate for our purposes in the original Lemm output.

**Unifying Synan and Lemm:** For every binary relation in Synan output, we take the corresponding parent and child analyses and find corresponding roles in the Lemm output, removing all other analyses. If the lemma for parent or child was null—as, e.g., when the corresponding element was a group—we infer information from Lemm output for the missing element. In cases when a word does not participate in any relation identified by Synan, its analysis is taken entirely from Lemm output, passing along any unresolved ambiguity.

**Table 3.** Result of the PULS AOT wrapper, combining morphology and syntax, and resolving ambiguity

| ID | Offset | Surface | Lemma | Relation | POS | Morphological tags |
|---|---|---|---|---|---|---|
| 0 | 0 | На | на | — | prep | |
| 1 | 3 | берегу | берег | prep-group→0 | noun | 2genl inan masc loc sg |
| 2 | 10 | пограничной | пограничный | adj-noun→3 | adj | inan anim fem gen sg |
| 3 | 22 | реки | река | gen-nom-group→1 | noun | inan fem gen sg |
| 4 | 27 | задержано | задержать | — | sparticip | past pass sg neut perf transv |
| 5 | 37 | двадцать | двадцать | card-ord-group→6 | card | nom |
| 6 | 46 | семь | семь | num-noun→7 | card | nom |
| 7 | 51 | нелегалов | нелегал | — | noun | anim masc gen pl |

**Output Generation:** After the unification, we assemble the resulting analyses for all words into a parse tree, or into a set of tree fragments. In some cases involving conjunctions, AOT produces two parents for a node; the wrapper adjusts the links so that they form a proper tree structure.

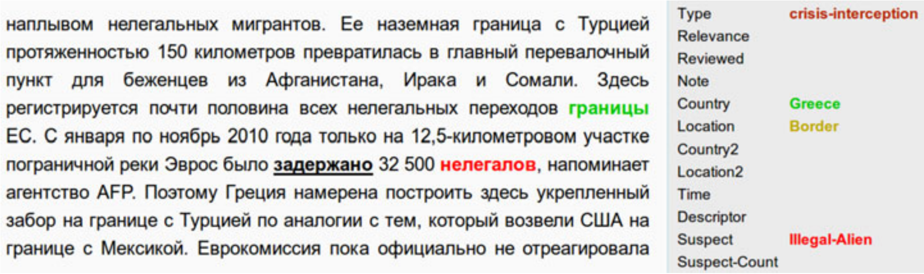
Table 3 shows the wrapper's output, which was modeled after Connexor parser output, [6]. This serves as the basis for the subsequent semantic analysis and IE.

## 5.2 Information Extraction

**Adapting the Ontology for Russian:** We use the existing PULS English-language ontology—including the domain-specific concepts—as the shared or *interlingua* concept base, and link directly to these concepts their instances in Russian (and other languages in the future). The base ontology needed to be extended in some cases, e.g., by making explicit certain concepts that may be ambiguous in English. For example, an English word (e.g., "arrest") can act as verb and noun, whereas in Russian they have different base forms—"арестовать" vs. "арест".

**Patterns and Inference Rules:** Patterns are used to group smaller syntactic units together into larger units, starting from the individual words in the sentence syntactically analyzed by AOT. For example, adjectives and numeric expressions are joined with the nouns they modify into noun phrases, genitive groups ("the house of the president of France") are joined into larger noun phrases, then into prepositional phrases, and so on. After the higher-level phrases are built from lower-level elements, domain-specific patterns and inference rules are applied to find events, based on semantic analysis, as follows.

We can construct an inference rule for finding arrest events in a sentence, e.g., of the kind *"perpetrator detained in location"*—for text like "преступник был задержан в Греции". The rule should fire if it finds phrases headed by concepts of type *perpetrator*, *arrest* and a locative prepositional phrase with *location*; further, the rule may specify that the concepts should be linked by a certain syntactic relation, or (more loosely) occur in the same sentence, or in nearby sentences. When the rule fires, it specifies which slots in the event are filled by which semantic constituents. An event template (partially) filled by such a rule is shown in Fig. 1, on a real-world news article. More advanced rules and patterns can exploit additional morphological and syntactic constraints.



наплывом нелегальных мигрантов. Ее наземная граница с Турцией протяженностью 150 километров превратилась в главный перевалочный пункт для беженцев из Афганистана, Ирака и Сомали. Здесь регистрируется почти половина всех нелегальных переходов **границы** ЕС. С января по ноябрь 2010 года только на 12,5-километровом участке пограничной реки Эврос было **задержано** 32 500 **нелегалов**, напоминает агентство AFP. Поэтому Греция намерена построить здесь укрепленный забор на границе с Турцией по аналогии с тем, который возвели США на границе с Мексикой. Еврокомиссия пока официально не отреагировала

| | |
|---|---|
| Type | crisis-interception |
| Relevance | |
| Reviewed | |
| Note | |
| Country | Greece |
| Location | Border |
| Country2 | |
| Location2 | |
| Time | |
| Descriptor | |
| Suspect | Illegal-Alien |
| Suspect-Count | |

**Fig. 1.** A sample document text, and a (partially) filled-in event template

## 6 Current Work

The framework we describe provides end-to-end functionality for Russian IE. Our current efforts center on improving performance, which mainly entails enriching the concept ontology and building patterns and rules. We pursue this via two approaches in parallel. First, we are adapting the patterns and rules in the pre-existing English-language system to Russian text, which involves a certain amount of manual labor. Second, this is aided by *bootstrapping* for lexicons and patterns, as tested previously on English and other languages, e.g., [15,13], which we are adapting for Russian. Lastly, further work is required on the AOT wrapper, as well as on extending the mechanisms for IE patterns and rules to utilize the full range of morphological and syntactic information provided by the lower-level analysis, which will help improve precision.

## References

1. Astaf'eva, I., Bonch-Osmolovskaya, A., Garejshina, A., Grishina, J., D'jachkov, V., Ionov, M., Koroleva, A., Kudrinsky, M., Lityagina, A., Luchina, E., Sidorova, E., Toldova, S., Lyashevskaya, O.S.S., Koval', S.: NLP evaluation: Russian morphological parsers. In: Proceedings of Dialog Conference, Moscow, Russia (2010)
2. Atkinson, M., Belyaeva, J., Zavarella, V., Piskorski, J., Huttunen, S., Vihavainen, A., Yangarber, R.: News mining for border security intelligence. In: Proceedings of IEEE ISI-2010: Intelligence and Security Informatics, Vancouver, BC, Canada (2010)
3. Atkinson, M., Piskorski, J., Tanev, H., van der Goot, E., Yangarber, R., Zavarella, V.: Automated event extraction in the domain of border security. In: Proceedings of MINUCS: Workshop on Mining User-Generated Content for Security, at the UCMedia: 1st International ICST Conference on User-Centric Media, Venice, Italy (2009)
4. Bontcheva, K., Maynard, D., Tablan, V., Cunningham, H.: GATE: A Unicode-based infrastructure supporting multilingual information extraction. In: Proceedings of Workshop on Information Extraction for Slavonic and other Central and Eastern European Languages, Borovets, Bulgaria (2003)
5. von Etter, P., Huttunen, S., Vihavainen, A., Vuorinen, M., Yangarber, R.: Assessment of utility in Web mining for the domain of public health. In: Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents, pp. 29–37. Association for Computational Linguistics, Los Angeles (June 2010), http://www.aclweb.org/anthology/W10-1105

6. Järvinen, T., Tapanainen, P.: A dependency parser for English. Tech. Rep. TR-1, Department of General Linguistics, University of Helsinki, Finland (February 1997)
7. Linge, J., Steinberger, R., Weber, T., Yangarber, R., van der Goot, E., Khudhairy, D.A., Stilianakis, N.: Internet surveillance systems for early alerting of health threats. Eurosurveillance Journal 14(13) (2009)
8. Piskorski, J., Atkinson, M., Belyaeva, J., Zavarella, V., Huttunen, S., Yangarber, R.: Real-time text mining in multilingual news for the creation of a pre-frontier intelligence picture. In: Proceedings of ISI-KDD: ACM SIGKDD Workshop on Intelligence and Security Informatics, at KDD-2010: 16th Conference on Knowledge Discovery and Data Mining, Washington, DC (2010)
9. Sokirko, A.: Semantic dictionaries in automatic text analysis, based on DIALING system materials. Ph.D. thesis, Russian State University for the Humanities, Moscow (2001)
10. Sokirko, A.: A short description of DIALING project (2001), `http://www.aot.ru/docs/sokirko/sokirko-candid-eng.html`
11. Sokirko, A.: Private communication (2011)
12. Steinberger, R., Fuart, F., van der Goot, E., Best, C., von Etter, P., Yangarber, R.: Text mining from the web for medical intelligence. In: Perrotta, D., Piskorski, J., Soulié-Fogelman, F., Steinberger, R. (eds.) Mining Massive Data Sets for Security, OIS Press, Amsterdam (2008)
13. Thelen, M., Riloff, E.: A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002 (2002)
14. Wilensky, R.: Common LISPcraft. W. W. Norton and Company, USA (1986)
15. Yangarber, R.: Counter-training in discovery of semantic patterns. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan (July 2003)

# Finding the Optimal Number of Clusters for Word Sense Disambiguation

Bartosz Broda and Paweł Kędzia

Institute of Informatics, Wrocław University of Technology, Poland
{bartosz.broda,pawel.kedzia}@pwr.wroc.pl

**Abstract.** Ambiguity is an inherent problem for many tasks in Natural Language Processing. Unsupervised and semi-supervised approaches to ambiguity resolution are appealing as they lower the cost of manual labour. Typically, those methods struggle with estimation of number of senses without supervision. This paper shows research on using stopping functions applied to clustering algorithms for estimation of number of senses. The experiments were performed for Polish and English. We found that estimation based on PK2 stopping functions is encouraging, but only when using coarse-grained distinctions between senses.

## 1 Introduction

Ambiguity of language is a stumbling block for many tasks in Natural Language Processing (hereafter NLP). Machine translation is an obvious example as there is a need to select the right sense of a word in the right context [1]. One of typical examples given in this context is the word *bank*, which can denote river bank or financial institution in a given context. The domain of *Word Sense Disambiguation* (WSD) deals with the problem of contextual resolution of ambiguities.

There are many approaches to WSD [1]. Among them methods using supervised Machine Learning (ML) achieve best precision of disambiguation in Senseval[1] competitions. Those approaches have a few drawbacks, e.g., the cost of manual preparation of resources is very high. The domain adaptation of systems is also a serious concern for WSD based on supervised ML. The annotation of corpora with word senses is not only laborious, but also prone to errors. Usually, annotation is being performed by more than one linguist, thus some differences in interpretation of word senses may arise.

To overcome those drawbacks one can employ approaches based on unsupervised or semi-supervised ML [1]. Typically, precision of disambiguation is lower for unsupervised and semi-supervised system in compassion to supervised ones. On the other hand, they require few (if any) hand-made resources. Usually those methods involve some form of clustering. There are some difficulties that need to be addressed before unsupervised methods will be applicable in practice. One of the problems is related to the required knowledge about the number of senses for a given word [2]. This is essential for clustering-based WSD as the quality of clustering strongly depends on the number of clusters. Using too few clusters results in merging of senses, but too many

[1] Sesneval is a competition devoted to evaluation of Word Sense Disambiguation systems and other semantic analysis tasks, see http://www.senseval.org/.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 388–394, 2011.

clusters can be harmful too, as it can lead to splitting of senses according to some subtle co-occurrence patterns.

The importance of the problem was noted in Senseval, i.e., last two competitions included *Word Sense Induction* task, where one of the difficulties arose from the need of finding the number of word senses without supervision. Moreover, other areas of NLP would benefit greatly if robust solution to the problem of estimation of word senses could be found. For example, building of lexicons would be easier if the lexicographer knew the number of words senses in advance.

There are a few methods that can be used to approach the problem of estimation of number of word senses. One example involve using statistical linguistic laws, e.g., Zipf's law or Menzerath-Altmann's law [3]. Those laws use very simple cues, e.g., word frequency or length. Results of our preliminary investigation with applying those (and other) statistical linguistic laws to the problem were not encouraging, i.e., we found that the laws describe the linguistic phenomena well, but their biggest drawback is the lack of predictive power.

There are a few clustering methods that try to find the right number of clusters automatically, e.g., [4]. Those approaches have a few drawbacks. For example, estimation of number of clusters is bound to the clustering methodology, so it is hard to apply them in situations where different clustering algorithms are better suited for a given task.

Last but not least, there are methods that use special *stopping functions* (or *stopping rules*) that help in estimation of the right number of clusters using any clustering algorithm [5]. In clustering-based unsupervised approaches to WSD finding the right number of clusters is equivalent to finding the number of senses. The general idea of those approaches is as follows: first the dataset is clustered into $1 \ldots k$ clusters, then the stopping rule scores each of the clustering solution; finally the solution with the optimal value of stopping function is used.

We focus on employing stopping functions in this paper due to their flexibility in determining the optimal number of words senses. Thus, the aim of this work is to investigate different stopping functions in the task of finding optimal number of clusters for WSD. We selected a few commonly used stopping rules in WSD literature [2]. We also include Calinski and Harabasz [6] stopping rule as it was found to perform the best among other 30 stopping functions in the extensive review of Millgian and Cooper [5]. We are mostly interested in processing Polish, but in order to gain better insights into the nature of tested stopping functions we also experiment with English datasets.

## 2 Selected Stopping Rules

Adapted GAP Statistic (AGS) [2] is an extension to GAP Statistic proposed in [7]. The difference is that [2] propose to use clustering *criterion function* (hereafter *crfun*) during both phases: clustering and selection of number of clusters. The basic idea of GAP is to compare changes in cluster tightness with the expected ones under a reference null distribution [7]. AGS starts with clustering of the dataset into $k$ clusters and counting within-cluster dispersion of each cluster $W_k$ using the following formula: $W_k = \sum_{l=1}^{k} \frac{1}{2n_l} D_l$, where $D_l$ is a sum of distances in cluster $l$ (using e.g., Euclidean distance). Afterwards, the null reference distribution is used to generate $B$ clusters and their within-cluster dispersion $W_k*$ is computed. The next step involves counting the $Gap(k) = (\frac{1}{B})\sum_{b=1}^{B}(\log(W_k*) - \log(W_k))$, which measures difference between real model and artificially generated data. To select the optimal number of cluster one need to minimize $gapK = min\{Gap(k) \geq Gap(k+1) + s_{k+1}\}$, where $s_k$ is modified standard deviation by the factor of $\sqrt{1 + \frac{1}{B}}$.

The *PK1* stopping function was first proposed in [8]. It was adapted by Pedersen and Kulkarni to the problem of WSD in a similar fashion to AGS (i.e., *crfun* used in clustering is used in estimation of number of clusters). This stopping function has the following form: $PK1(k) = \frac{crfun(k) - mean(crfun[1...maxK])}{std(crfun[1...maxK])}$. One need to select appropriate threshold for PK1 value. Value of PK1 exceeding the threshold indicate optimal number of clusters. [2] found that the optimal value of threshold is $-0.7$ for WSD data. [8] used values from the interval $< 2.75, ..., 3.5 >$. In our experiments we found that $0.5$ is optimal.

*PK2* stopping function compares *crfun* for two subsequent values of $k$, i.e., $PK2(k) = \frac{crfun(k)}{crfun(k-1)}$. The optimal number of clusters is selected when PK2 is closest to 1 (but not below 1). [2] use standard deviation of PK2 to find the optimal value of PK2.

*PK3* uses three subsequent values of $k$ to determine optimal number of clusters. It combines them in a similar manner to Dice Coefficient, but uses *crfun*. Namely, it uses the following formula: $PK3(k) = \frac{2*crfun(k)}{crfun(k-1)+crfun(k+1)}$. The PK3 is run for different values of $k$ until PK3 value is greater than its standard deviation.

Calinski and Harabasz proposed a stopping function that uses both within- and between-cluster distances [6]. It uses the following formula: $CH = \frac{\frac{BGSS}{(k-1)}}{\frac{WGSS}{n-k}}$, where $BGSS$ is a sum of squares of between-cluster pair-wise distances, $WGSS$ denotes sum of squares of within-cluster cluster distances, $n$ is the size of the dataset. Quadratic Euclidean distance is usually used with CH [6].

## 3 Datasets

We are mostly interested in WSD for Polish, thus we start with recently developed corpus for Polish called SCWSD[2] [9]. SCWSD contains 1344 annotated contexts[3] of 13 polysemous nouns. Typically for language usage the sense distribution is skewed. There are 72 word senses annotated, among which 8 occur only once, 13 with frequency 2–5, 27 frequency 6–19 and 24 occur in the corpus more frequently. SCWSD was annotated with senses taken from the Polish wordnet[4] called plWordNet [11]. Wordnets strive for including all senses, with fine-grained sense distinctions. It is unrealistic to require that all senses for a given set of words will occur in a given corpus (e.g., the words annotated in SCWSD have 101 senses, but only 72 were found and annotated in the corpus constructed with a sense completeness in mind, see [9]). Also, for some applications disambiguation of closely related senses might not be necessary. For example, in machine translation one need to disambiguate between senses that have different translations in

[2] Polish name: *Korpusik US II PWr*, rough translation: Small Corpus for WSD.

[3] By a *context* we mean a short passage of text containing a polysemous word.

[4] A wordnet is an electronic thesaurus constructed in the way similar to Princeton WordNet [10].

a target language. Thus, we manually created a grouping of the senses in SCWSD. We will call this dataset SCWSD-coarse.

The work on manually annotated corpus is interesting, as we know exactly the number of senses that occur in texts. On the other hand, we would like to estimate the number of senses on the basis of large unannotated corpora, because large amount of data increase probability of finding greater number of senses. Thus, we construct two test datasets from large corpora. First one contains all occurrences of 13 nouns found in SCWSD gathered from three corpora: the IPI PAN Corpus (IPIC) [12], Rzeczpospolita newspaper [13] and set of large documents collected from the Web. The joint corpus contains roughly 500 million words. We will use all the senses from plWordNet (we will call this dataset LC-13) as well as the sense groupings mentioned earlier (LC-13-coarse). The 13 words in SCWSD were selected on the basis of early version of plWordNet, thus they include only high-frequency nouns occurring in IPIC (i.e., the least frequency word occurs 533 times). To supplement the data we use another set of 33 polysemous nouns. The words were carefully chosen with the help of two professional linguists in order to include as much polysemy-related phenomena as possible. We used only their occurrences in IPIC for estimation of number of senses, so we call this dataset IPIC-33.

In order to gain a better insight into the nature of investigated stopping functions we also use datasets from first four English *lexical sample*[5] tasks from Senseval competitions [14,15,16,17]. Senseval datasets are more diverse than those used for Polish. They include verbs and adjectives and words with only one sense annotated. Also, there are dissimilarities among the Senseval datasets in regard to the sense distinctions, corpora used and annotation process.

## 4 Experiments

We perform all the experiments using repeated bisection clustering algorithm as it was successfully applied to WSD [2] and other NLP tasks [18,11]. We apply $e1$ and $i1$ criterion functions [18]. In order to cluster text snippets one need to extract discriminating features. There are many possible features one can use in WSD [1]. In this work we employ only simple unigram features, i.e., bag-of-words representation. Thus, every context of ambiguous word is represented as a vector in a high-dimensional space, where every dimension corresponds to one of the words occurring in the contexts. We use cosine as a similarity measure during the clustering.

We do not expect that any stopping function will be able to find the optimal number of senses. There are two reasons for that: not all the senses are present in any corpus and sense distribution is skewed (so clustering algorithm will struggle with finding infrequent senses). Also, small errors in estimated number of senses might not be a problem in certain NLP tasks. However, not all mistakes are equal, i.e., missing one sense of a highly polysemous word (e.g., 1 of 30 nominal senses of *line*) is less of a problem than missing one sense of a word that has only a few senses (e.g., 1 of 3 senses of *onion*). Thus, we count the number of *acceptable* answers of stopping function in the following way: for words with less than 4 senses we accept answers with one additional

[5] Lexical sample task deals with disambiguation of only selected words.

**Table 1.** Precision of selected stopping function in discovering number of word senses

| Dataset | PK1 [%] | PK2 [%] | PK3 [%] | CH [%] | AGS [%] |
|---|---|---|---|---|---|
| SCWSD | **69.23** | 46.15 | 7.69 | 7.69 | 7.69 |
| SCWSD-coarse | 46.15 | **76.92** | 38.46 | 30.77 | 23.08 |
| LC-13 | **15.38** | **15.38** | 0.00 | 0.00 | 0.00 |
| LC-13-coarse | 53.85 | **76.92** | 30.77 | 15.38 | 38.46 |
| IPIC-33 | 6.06 | **54.54** | 15.16 | 30.30 | 15.16 |
| Senseval-1 | 11.11 | **22.22** | 14.81 | 11.11 | 16.67 |
| Senseval-2 | 4.10 | **31.50** | 1.37 | 2.73 | 9.59 |
| Senseval-3 | 1.75 | **17.54** | 1.75 | 0.00 | 1.75 |
| Senseval-4 | 3.00 | **62.00** | 34.00 | 12.00 | 29.00 |

sense too; for words that have less than 10 senses we allow mistakes by ±1 sense; and for more polysemous words we allow mistakes by ±2 senses. We calculate precision as a ratio of acceptable answers to number of words. The evaluation methodology is easy to interpret as opposed to the one used in, e.g., [2,19], but unfortunately the results are not directly comparable.

All the stopping function requires repeated clustering with different number of senses, so the amount of computing power for performing the experiments is significant.[6]

We started the experiments with investigation of the impact of dimensionality reduction on the precision. We use Singular Value Decomposition (SVD) to reduce the dimensionality to 10% of the original dimension for words that are described by more than 30 000 dimensions, and to 70% for words described by more then 1000. As expected, SVD resulted in improvement of results by 3% on average.

Table 1 summarises the precision of selected stopping function on all the datasets used.[7] Among all the stopping function PK1 and PK2 achieved best precision. The low results for Calinski and Harabasz (CH) are surprising as it was found to be the best among 30 others stopping functions in [5]. Also, low precision of AGS is unexpected as this method has sound theoretical foundations [7]. This might be caused by the usage of Euclidean distance in both CH and AGS. SVD brought some improvements, but using other distance measure that is more robust for high-dimensional data might be necessary.

The results for fine-grained sense distinctions are not encouraging. This is caused by two factors. For LC-13 not all senses from plWordNet are present in corpora. Also, many senses have only a few occurrences, thus clustering algorithms are not able to form groups describing those senses. On the other hand, results for coarse-grained sense distinctions are encouraging. Results for IPIC-33 are mixed, as the dataset include both highly polysemous words and words with only a few senses.

[6] Also, randomness is involved (e.g., during generation of reference model for AGS), thus we repeated all the experiments 10 times. Our experiments took about three weeks using 30 cores (Intel Xeon at 2.67 GHz and Intel Core i7 at 2.80 GHz).

[7] We report only results using SVD for brevity.

We investigated the influence of the reference dictionary used for evaluation. We used online edition of Słownik Języka Polskiego PWN (SJP)[8] as additional gold standard for LC-13. In comparison with plWordNet the precision for both PK1 (up to 46.15%) and PK2 (up to 23.08%) was higher. Similarly for IPIC-33 the precision for both PK1 (up to 12.12%) and PK2 (up to 63.63%) was higher when compared with the number of senses from SJP. This might indicate that senses in plWordNet are too fine-grained for finding them in a fully automatic way in corpora.

Results for English are quite low, but consistent with those achieved in [2]. Those results are also consistent with the results for Polish. Namely, PK2 is the most precise stopping function and for coarse-grained senses (Senseval-4, [17]) the results are satisfactory. On the other hand, grouping of word senses in Senseval-3 (cf [16]) does not improve the results significantly. This might be caused by the fact, that the corpus was annotated by Web users, so the quality of annotations is low.

## 5 Conclusions and Further Works

This paper shows research on using clustering algorithms for estimation of the number of senses. Experiments were performed using two languages and a few different dictionaries. We found that PK2 stopping function achieves the best precision in majority of cases. We also found that none of the tested stopping function is able to determine the number of senses reliably while using a dictionary with fine-grained sense distinctions.

The results for coarse-grained sense distinction are encouraging. This work further supports previous observations that usage of fine-grained senses might be inappropriate for automatic WSD as humans are not able to achieve high inter-annotator agreement [20,17].

In the future we plan to extend CH and AGS stopping functions with a distance function more suitable for high-dimensional data. Extraction of other types of features and using second order co-occurrence might also help. We plan to investigate other dimensionality reduction techniques. In this work we focused only on finding the right number of senses in an automatic way. It remains to be shown how tested stopping functions influence the process of unsupervised and semi-supervised word sense disambiguation.

**Acknowledgments.** Work financed by Innovative Economy Programme project POIG.01.01.02-14-013/09. We would like to thank Justyna Ławniczak and Marek Maziarz for help in selection of words used in IPIC-33 dataset.

## References

1. Agirre, E., Edmonds, P. (eds.): Word Sense Disambiguation: Algorithms and Applications. Springer, Heidelberg (2006)
2. Pedersen, T., Kulkarni, A.: Selecting the right number of senses based on clustering criterion functions (2006)
3. Pawlowski, A.: Metody kwantytatywne w sekwencyjnej analizie danych. English title: Quantitative methods in sequential data analysis. Katedra Lingwistyki Formalnej Uniwersytetu Warszawskiego (2006)

[8] http://sjp.pwn.pl/

4. Frey, B., Dueck, D.: Clustering by passing messages between data points. Science 315(5814) (2007)
5. Milligan, G., Cooper, M.: An examination of procedures for determining the number of clusters in a data set. Psychometrika 50(2), 159–179 (1985)
6. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics Simulation and Computation 3(1), 1–27 (1974)
7. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal Of The Royal Statistical Society Series B 63(2), 411–423 (2001)
8. Mojena, R.: Hierarchical grouping methods and stopping rules: an evaluation. Computer Journal 20(4) (1977)
9. Broda, B., Piasecki, M., Maziarz, M.: Evaluating LexCSD — a weakly-supervised method on improved semantically annotated corpus in a large scale experiment. In: Intelligent Information Systems (2010)
10. Fellbaum, C., et al.: WordNet: An electronic lexical database. MIT press, Cambridge (1998)
11. Piasecki, M., Szpakowicz, S., Broda, B.: A wordnet from the ground up. Oficyna wydawnicza Politechniki Wroclawskiej (2009)
12. Przepiórkowski, A.: The IPI PAN Corpus: Preliminary version. Institute of Computer Science PAS (2004)
13. Weiss, D.: Korpus Rzeczpospolitej (2008), http://www.cs.put.poznan.pl/dweiss/rzeczpospolita
14. Kilgarriff, A., Rosenzweig, J.: Framework and results for English SENSEVAL. Computers and the Humanities 34(1), 15–48 (2000)
15. Edmonds, P.: SENSEVAL: The evaluation of word sense disambiguation systems. ELRA Newsletter 7(3), 5–14 (2002)
16. Mihalcea, R., Chklovski, T., Kilgarriff, A.: The Senseval-3 English lexical sample task. In: 3rd Int. Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pp. 25–28 (2004)
17. Pradhan, S.S., Loper, E., Dligach, D., Palmer, M.: SemEval-2007 task 17: English lexical sample, SRL and all words. In: Proc. of the 4th International Workshop on Semantic Evaluations, pp. 87–92. ACL (2007)
18. Zhao, Y., Karypis, G.: Empirical and theoretical comparisons of selected criterion functions for document clustering. Machine Learning 55(3), 311–331 (2004)
19. Pedersen, T., Kulkarni, A.: Automatic cluster stopping with criterion functions and the Gap Statistic. In: Proceedings of the Demo Session of NAACL (2006)
20. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: the 90% solution. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX, pp. 57–60. Association for Computational Linguistics (2006)

# hrWaC and slWac: Compiling Web Corpora for Croatian and Slovene

Nikola Ljubešić[1] and Tomaž Erjavec[2]

[1] Faculty of Humanities and Social Sciences, University of Zagreb, Croatia
nikola.ljubesic@ffzg.hr
[2] Dept. of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
tomaz.erjavec@ijs.si

**Abstract.** Web corpora have become an attractive source of linguistic content, yet are for many languages still not available. This paper introduces two new annotated web corpora: the Croatian hrWaC and the Slovene slWaC. Both were built using a modified standard "Web as Corpus" pipeline having in mind the limited amount of available web data. The modifications are described in the paper, focusing on the content extraction from HTML pages, which combines high precision of extracted language content with a decent recall. The paper also investigates text-types of the acquired corpora using topic modeling, comparing the two corpora among themselves and with ukWaC.

**Keywords:** web corpus, Croatian, Slovene, topic modeling.

## 1 Introduction

With the advent of the web, a vast new source of linguistic information has emerged. The exploitation of this resource has especially gained momentum with the WaCky initiative [1], which has popularised the concept of ”Web as Corpus”. It has also made available tools for compiling such corpora and produced large WaC corpora for a number of major European languages. Now such corpora are also being built for the so called smaller languages, such as Norwegian [2] and Czech [3], moving the concept of a ”large corpus” for smaller languages up to the 1 billion token frontier. As Web corpus acquisition is much less controlled than that for traditional corpora, the necessity of analyzing their content gains in significance. The linguistic quality of the content is mostly explored through word lists and collocates [1] while the content itself is explored using unsupervised methods, such as clustering and topic modeling [4].

## 2 Building the hrWaC and slWaC

The standard pipeline for building web corpora was developed primarily for languages where the amount of web data is orders of magnitude larger than the corpus being built. On the other hand, smaller languages cannot afford the luxury of sampling since the amount of data is limited - alternatively, this can be seen as a bonus, as a large

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 395–402, 2011.

portion of all the language Web can be turned into a language corpus[1]. In this paper we propose a modified traditional pipeline which would better suit smaller languages with a limited amount of web data available. Additionally, we describe a novel content extraction method with high precision and a decent recall.

A detailed overview of the numerical results of applying the pipeline for the two corpora is given in Table 1.

**Table 1.** Numerical summary of the corpus creation process. This version of the corpora was crawled between January and March, 2011.

| | hrWaC | slWaC |
|---|---:|---:|
| # of seed domains | 12,033 | 11,493 |
| # of domains crawled | 16,398 | 18,418 |
| # of crawled documents | 15,747,585 | 9,247,341 |
| # of documents after deduplication | 14,654,394 | 9,022,716 |
| # of extracted documents | 3,924,194 | 1,598,011 |
| # of language identified documents | 3,607,054 | 1,337,286 |
| # of non-filtered documents | 3,409,226 | 1,287,895 |
| # of tokens | 1,186,795,086 | 380,299,844 |

### 2.1 Collecting Seeds, Crawling, Physical Deduplication

For collecting seed URLs we used the Yahoo! Search BOSS API. The Yahoo search index was queried with random bigrams composed of mid-frequency tokens (frequency rank 1,000-10,000 from 100-million-token newspaper corpora) and about 50,000 URLs were collected for every language. Since Croatian and Slovene webs are much smaller than those for ”large” languages an early decision was made not to sample the web as in the case of English, German or French, but to crawl it completely. For that reason only top pages of the collected domains on the *hr* and *si* top domains were used as seeds for the crawling process.

Crawling was performed with a multi-threaded, breadth-first crawler developed for this purpose since most available crawlers lack precise control, such as filtering by the MIME file type. Only ”text/html” files of size between 50 and 500 kB were crawled. Both crawls ran several weeks and collected 15.7 million Croatian and 9.2 million Slovene documents.

The next step in the pipeline was physical deduplication, i.e. removing all but one copy of files that are physically identical. For that task the SHA224 hashing algorithm was used. During the process 2.25% of the Croatian documents and 2.3% of the Slovene ones were removed.

[1] Estimating the number of documents per language via Google (five most frequent words with language filter) on 2011-05-27 yields these numbers: English 25.27 billion (4.7 billion on *uk* domain), German 2.27 billion (1.27 billion on *de* domain), Norwegian 332 million (261 million on *no* domain), Croatian 229 million (70 million on *hr* domain), Slovene 210 million (82 million on *si* domain).

## 2.2 Content Extraction

A crucial step in building a web corpus is the content extraction step, often called boilerplate removal. We prefer the first term since it is our belief that just a part of the HTML document should be retained, rather than just a portion of the document removed.

This processing step has undergone the most changes in contrast to the classic WaCky pipeline. We did not use the well known body text extraction (BTE) algorithm, but a novel, more conservative algorithm, which aims at a very high precision, but without too great penalties on the recall. In our opinion this is the phase where most noise enters the corpus, which can have negative impact not only on the linguistic quality of the corpus, but also indirectly on its size. An example of boilerplate removal coupled with near-duplicate removal possibly gone bad is the Norwegian web corpus noWaC [2] where on the near-duplicate removal step 90% of documents were removed, probably because of boilerplate remains, which then identified many documents as near-duplicates.

Our algorithm is based on the notion that the largest amount of linguistically relevant content can be found by identifying the largest chunk of graphically identical and linguistically correct formatted text in the document. Since almost all web sites nowadays use CSS for styling and define the CSS class used in *id* and *class* attributes of the HTML elements, it was our assumption when building the algorithm that by identifying the largest amount of text on the same depth in the DOM node tree with same formatting we will identify the document body. Therefore our rather simple algorithm proceeds as follows:

1. Pre-format the HTML document by enclosing all text divided by *br* elements into separate paragraph elements
2. Represent every paragraph node in the DOM node tree as the path of triples (*tag*, *id attribute*, *class attribute*) from the root node down to the node of interest[2]
3. For every paragraph element which satisfies the constraints of well formatted paragraphs defined by a series of regular expressions calculate its weight via the formula

   $$weigth(p) = \frac{text\_length(p) * (1 - link\_density(p))}{size(map_{tic:w})} \tag{1}$$

   where the $text\_length()$ function returns the number of characters in the paragraph, the $link\_density()$ function returns the percentage of characters being part of a link and the $size(map_{tic:w})$ expression returns the number of elements in the map where the sum of weights of elements with specific tag-id-class paths is stored.
4. Add the calculated weight to the $map_{tic:w}$ map under the corresponding tag-id-class path.
5. Choose the tag-id-class path with the maximum weight and return the textual content of all previously analyzed elements having that tag-id-class path.

By multiplying the text length with the percentage of the not-linked text we take into account only the amount of ”clean” text while by dividing it further with the number

[2] An actual example of such path is *((div, container, ), (div, wrap, ), (div, content, content_article), (div, article_text, article_text))*. Only the nodes below the *body* node are recorded since higher nodes are constant.

of different tag-id-class paths found up to that moment, larger weights are given to the elements found sooner while traversing the DOM tree. The weighting function coupled with the constraints of well formatted paragraphs follows the intuition that the main text of a document will be the largest amount of linguistically well formatted text not containing many links which is found rather at the beginning of the document. Same graphical formatting is ensured by the uniformity of the tag-id-class path.

An evaluation of the algorithm was performed on 200 online newspaper documents downloaded from 20 different news portals.[3] As competing methods we chose the BTE algorithm, due to its heavy usage in the WaCky community as implemented in BootCaT[4] and the BoilerPipe 1.1.0 API,[5] due to its recent popularity in the NLP/IR community. In the experiment we call our algorithm ContentExtractor. The precision and recall evaluation measures were calculated via LCS (longest common subsequence) where the result was normalized for precision by the length of the extracted text while for recall the result was normalized by the length of the gold standard. The results of the experiment are given in Table 2.

**Table 2.** Precision, recall and F1 of three different algorithms on the task of content extraction

| | precision | recall | F1 |
|---|---|---|---|
| ContentExtractor | 0.979 | 0.707 | 0.821 |
| BTE | 0.570 | 0.955 | 0.713 |
| BoilerPipe | 0.847 | 0.921 | 0.882 |

The results show an best overall performance of the BoilerPipe algorithm. On the other hand, BTE has an even greater recall than BoilerPipe but with a drastic drop in precision. The distinction of ContentExtractor is a very high precision, but with lower recall. Obviously, each of the algorithms has its advantages. If one was aiming at high recall, being ready to clean up the result in later stages, BTE could be a good solution. A downside of the implementation of the BTE algorithm in BootCat is that it looses all structural information, and therefore makes latter clean-up very complicated. On the other hand, the other two algorithms keep the paragraph structure intact. If one needed a middle approach with both decent precision and recall, the BoilerPipe algorithm would be the best choice. It is our belief that collecting HTML data primarily for the purpose of modeling linguistic phenomena, loosing some text elements like titles, headings and lists is a tolerable (if not desirable?) loss. The omission of these text structures enables very high precision; also, today's corpus investigations seldom cross paragraph boundaries. One could wonder on this stage why we chose an algorithm with lower recall having in mind the smaller amount of available data in the first place. It is our belief

[3] An implementation of the algorithm and the evaluation sample are published on `http://www.nljubesic.net/hrWaC_Croatian_Web_Corpus/content_extraction.html`

[4] The PotaModule module was obtained from `http://bootcat.sslmit.unibo.it/`

[5] The ArticleExtractor class optimized for newspaper articles was used `http://code.google.com/p/boilerpipe/`

that this is an inevitable data loss if one wanted to obtain a clean and thereby usable resource.

Since we made an early decision to avoid the step of near duplicate removal because of (I) the danger of loosing a large amount of data on false positives (II) our belief that the problem of repeating content on smaller webs is much smaller than on the webs of larger languages and (III) its overall complexity, we bundled an additional step of duplicate removal with content extraction by removing identical paragraphs on the level of each domain.

By using the developed algorithm to extract text from the crawled documents for hrWaC and slWaC the conservativeness of our approach is shown by the fact that ContentExtractor returned text from only 26.5% of Croatian and 17.8% of the Slovene documents.

### 2.3 Language Identification, Filtering and PoS Tagging

After extracting linguistic content from HTML documents, we performed language identification with a combination of a second-order Markov chain model and a function word filter for 22 languages. We lowered the level of language identification to the paragraph level since our research showed that the error rate on paragraph level by combining two classifiers in a smart fashion is the same as with second-order Markov chain models on the document level [6]. Through the language identification step we experienced an approximately 8% document loss in Croatian and 17% document loss in Slovene. The higher loss in the Slovene corpus could be due to its membership in the European union and the consequent larger number of documents on the Slovene domain written primarily in English.

Additional filtering was performed to eliminate too short documents, those with encoding errors and those with a high amount of punctuation (often no running text like lists, document abstracts etc.) At the document filtering step 5.4% of Croatian and 3.7% of Slovene documents were removed.

The Croatian corpus was PoS-tagged and lemmatised with the tagger developed in the Institute for Linguistics at the Faculty of Humanities and Social Sciences, University of Zagreb [7], while the Slovene corpus was tagged and lemmatised with ToTaLe [8] trained on JOS corpus data [9]. The two taggers share harmonised PoS or, better, MSD (morphosyntactic description) tagsets, as both follow the MULTEXT-East morphosyntactic specifications [10].

As shown in Table 1, the final number of tokens is 1.2 billion for hrWaC and 380 million for slWaC. However, this is just a first version of the two corpora and our intention is to continue collecting new data with the primary goal of enhancing the size of the Slovene corpus.

## 3 Corpus Comparison

In this section we explore the content of the web corpora through the topic modeling method already used for corpus analysis tasks [4]. Our models were built with MALLET, [11] used with the default settings.

**Table 3.** Twenty hrWaC and slWaC topics with the amount of text they cover and up to ten words with highest probability. Topic names in bold are present in both hrWaC and slWaC.

| Lg | Topic name | Size | Words with highest probability |
|---|---|---|---|
| sl | intl. politics | 4.7% | leto država vojna človek predsednik oblast zda napad vojska dan |
| hr | **reg. politics** | 5.9% | zemlja srbija predsjednik godina država rat vlada hrvatska |
| sl | **reg. politics** | 3.6% | država slovenija eu leto članica minister predsednik hrvaška |
| hr | **dom. politics** | 6.2% | predsjednik vlada stranka izbor sanader ministar pitanje zakon |
| sl | **dom. politics** | 4.9% | vlada predsednik zakon stranka slovenija minister sodišče |
| hr | **law** | 3.0% | zakon tema odluka pravo postupak sud članak osoba ugovor |
| sl | **law** | 4.4% | zakon podatek primer pravica člen oseba plačilo storitev dan k |
| hr | crime | 5.3% | policija sud godina osoba slučaj zatvor kazna sat policajac sudac |
| hr | **finance** | 7.1% | godina kuna milijun tvrtka cijena banka euro tržište dionica |
| sl | **finance** | 5.2% | leto evro odstotek podjetje milijon družba banka cena trg država |
| hr | **sports** | 2.0% | utrka mjesto godina natjecanje prvenstvo vrijeme sezona staza |
| sl | **sports** | 4.0% | tekma minuta igra leto točka prvenstvo igralec ekipa mesto |
| hr | soccer | 4.8% | utakmica igrač klub momčad minuta pobjeda liga sezona trener |
| sl | classified ads | 2.9% | oglas iskanje seznam stran znamka stroj možnost vrh kvadrat |
| sl | environment | 6.0% | voda energija prostor barva material sistem uporaba del naprava |
| hr | **automoto** | 3.8% | motor automobil vozilo model boja auto sustav dio energija |
| sl | **automoto** | 2.6% | vozilo avtomobil motor dirka vožnja leto voznik avto kolo mesto |
| hr | web | 4.8% | dan godina stranica članak web informacija rubrika hr internet |
| hr | **IT** | 4.2% | korisnik internet uređaj igra stranica slika računalo program |
| sl | **IT** | 6.1% | stran uporabnik podatek sistem računalnik slika program |
| hr | construction | 5.5% | grad područje cesta dio voda godina stan kuća zgrada prostor |
| hr | **local themes** | 7.0% | godina grad županija dan škola sat udruga zagreb izložba rad |
| sl | **local themes** | 5.4% | občina leto cesta mesto prostor ljubljana članek območje |
| hr | **education** | 6.4% | rad godina projekt program škola razvoj sustav područje student |
| sl | **education** | 8.0% | delo področje program projekt leto šola razvoj organizacija |
| hr | **health** | 4.2% | bolest dan hrana voda godina koža lijek liječnik tijelo ulje |
| sl | **health** | 3.6% | telo bolezen zdravilo zdravnik leto otrok bolnik zdravljenje |
| hr | **travel** | 3.8% | hotel brod sat more mjesto otok grad godina dan soba |
| sl | **travel** | 5.3% | mesto dan pot ura leto hotel morje otok čas soba |
| hr | **family** | 9.3% | čovjek dan vrijeme mi život žena put djeca stvar godina |
| sl | **family** | 10.8% | človek čas življenje stvar svet ženska otrok dan način moški |
| hr | **religion** | 3.7% | čovjek crkva život bog svijet knjiga vrijeme godina riječ djelo |
| sl | **religion** | 3.2% | otrok leto cerkev dan oče bog človek čas mati roka |
| hr | forum | 3.1% | mi čovjek stvar dan vec jel problem par kajati godina |
| sl | lifestyle | 4.0% | koža hrana voda olje žival pes rastlina izdelek vrsta las |
| sl | art | 4.4% | leto knjiga delo razstava ljubljana avtor umetnost jezik zbirka |
| hr | **movies** | 6.2% | film godina svijet priča uloga glumica žena glumac serija život |
| sl | **movies** | 5.5% | film leto vloga igralec režiser zgodba nagrada svet igralka serija |
| hr | **music** | 3.7% | pjesma album koncert godina glazba festival publika predstava |
| sl | **music** | 5.4% | leto skupina glasba pesem koncert festival album dan oddaja |

The documents used for training the topic models were a 10% random sample of the corpus documents with their content stripped down to noun lemmata. Experimenting with the amount of data necessary for constant resuts showed that modeling on 1/10 of randomly chosen data does not change the results significantly, but, of course, does speed up this computationally demanding task significantly.

The number of the topics was set to 20, and the topics were manually named by examining the topic keywords. The resulting topics are shown with their size, counted as the number of tokens, and up to ten most probable terms in Table 3 for the two corpora. As can be seen, the two topic modeling results are quite similar. Fifteen out of twenty topics can be considered almost identical. The more prominent topics on the Croatian side are crime, soccer, the web, construction and on-line forums, while the topics prominent for Slovene are international politics, classified ads, environment, lifestyle and art.

When we compare these topic results to the results obtained from ukWaC with the same method, the similarity still remains high, but lower than between hrWaC and slWac. ukWac shares 13 similar topics with both, an additional one with slWaC and two with hrWaC.

In other words, (European) web corpora are rather similar to each other, regardless of the language they are produced in, nevertheless showing greater similarity between web corpora of culturally and linguistically more related languages.

## 4 Conclusion

The paper presented two new Web corpora, for Croatian and Slovene, and the pipeline that was used for building them. The pipeline introduces some changes to the current methods for Web corpus building, especially in the crucial step of content extraction, leading to cleaner corpora. Since the amount of available information in corresponding languages is not as high as for other, larger languages, our method manages to bypass methods known for eliminating a considerable amount of collected data. Further work is necessary to compare the quality of these corpora compared to already existing Web corpora.

Additionally, we analyzed the content of the built corpora via topic modeling and compared them to the ukWaC corpus showing a very high degree of similarity between the Web corpora of linguistically and culturally near languages and an overall high degree of similarity between Web corpora in general.

Further work includes, in the first place, enlarging the Slovene part of the corpus, which now lags behind its Croatian counterpart. The corpora will also be made available via a concordancer, possibly via SketchEngine. The main opportunity, however, for these corpora lies in using them for a series of modeling and extraction tasks, like the one currently underway - building comparable corpora of closely related languages for bilingual lexicon extraction.

## References

1. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation 43(3), 209–226 (2009)
2. Guevara, E.: NoWaC: a large web-based corpus for Norwegian. In: NAACL HLT 2010 Sixth Web as Corpus Workshop, pp. 1–7 (2010)
3. Spoustov, D., Spousta, M., Pecina, P.: Building a Web Corpus of Czech. In: Seventh Intl. Conf. on Language Resources and Evaluation, LREC 2010 (2010)
4. Sharoff, S.: Analysing Similarities and Differences between Corpora. In: 7th Conference ”Language Technologies”, Jožef Stefan Institute, Ljubljana, pp. 5–11 (2010)
5. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: WSDM 2010, pp. 441–450 (2010)
6. Stupar, M., Jurić, T., Ljubešić, N.: Language Identification on Web Data for Building Linguistic Corpora. In: Proceedings of the INFuture 2011 Conference (2011) (in press)
7. Agić, Ž., Tadić, M.: Evaluating Morphosyntactic Tagging of Croatian Texts. In: Fifth Intl. Conf. on Language Resources and Evaluation (2006)

8. Erjavec, T., Ignat, C., Pouliquen, B., Steinberger, R.: Massive Multilingual Corpus Compilation: Acquis Communautaire and ToTaLe. Archives of Control Sciences 15(3), 253–264 (2005)
9. Erjavec, T., Krek, S.: The JOS morphosyntactically tagged corpus of Slovene. In: Sixth Intl. Conf. on Language Resources and Evaluation (2008)
10. Erjavec, T.: MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In: Seventh Intl. Conf. on Language Resources and Evaluation (2010)
11. McCallum, A.K.: MALLET: A Machine Learning for Language Toolkit (2002), `http://mallet.cs.umass.edu`

# Question Classification for a Croatian QA System

Tomislav Lombarović, Jan Šnajder, and Bojana Dalbelo Bašić

Faculty of Electrical Engineering and Computing,
University of Zagreb, Croatia
{tomislav.lombarovic,jan.snajder,bojana.dalbelo}@fer.hr

**Abstract.** Question Answering (QA) systems provide efficient means for retrieval of information, which in many cases more directly address users' information needs. The performance of a QA system crucially depends on its ability to correctly classify the query question according to the expected answer type. This paper addresses the problem of a question classification for the Croatian language, as a first step towards building an open-domain QA system. We compare different machine learning classifiers on a Croatian test collection based on a two-level question taxonomy. The evaluation results are encouraging and comparable to state-of-the-art results for other languages: accuracy is over 80% for coarse-grained classification and almost 70% for fine-grained classification.

**Keywords:** Question classification, question answering, information retrieval, Croatian language.

## 1 Introduction

Large amounts of information are available today and its volume constantly increases. Therefore, the need for effective search is evident and it is necessary to explore means for retrieving targeted information. Question answering systems (QA) enable highly targeted information retrieval by providing the user with answers to questions written in natural language, unlike traditional search engines that return a ranked list of documents relevant to a keyword query. Research in QA began already in the 1960's when first closed-domain QA systems BASEBALL [3] and LUNAR [17] were developed. Since then there has been a steady increase in research, boosted by the evaluation campaigns such as TREC QA track [13] and CLEF multilingual QA track [7]. Traditionally most research has focused on English. More recently there has been work on QA for Slavic languages, such as Bulgarian [12], Polish [15], and Slovene [1].

Processing of questions in natural language is not an easy task and must be broken down into several steps. These are typically question classification, document retrieval, paragraph of passage retrieval, and – optionally – answer synthesis. First step is the classification of questions according to the expected answer type. If a QA system "knows" the type of the answer it is looking for (e.g., number, city, person name), it can apply the most efficient search and extraction strategy for the given answer type. For example, the question *"How many countries border with the Czech Republic?"* would be classified as NUMERIC-COUNT, and the question *"What countries border with the Czech Republic?"* would be classified as LOCATION-COUNTRY. The answers to these questions do

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 403–410, 2011.

not contain the phrases *"How many"* nor *"What"*, which happen to be the only keywords by which these two questions differ. However, by knowing the expected answer type, a QA system should be able to extract the correct answer to both questions.

The focus of this work is on question classification (QC) task as the first step towards building an open-domain QA system for the Croatian language. To the best of our knowledge, this is the first work on question classification for Croatian. In terms of question classification, one notable difference between English and Croatian is the fact that Croatian – like all Slavic languages – is morphologically more complex, which makes classification a more difficult task. In this work we report on experiments with four classification methods: three classical machine learning methods and classification based on statistical language modeling. The classifiers are evaluated on a Croatian QC test collection based on a two-level question taxonomy.

The rest of the paper is organized as follows. Next section briefly describes the related work on question classification. In Section 3 we describe question type classification for the Croatian language, while in Section 4 we describe the Croatian QC test collection. Results are presented and discussed in Section 5. Section 6 concludes the paper and outlines future work.

## 2 Related Work on Question Classification

The QC problem can be tackled using various approaches, ranging from rule-based methods (e.g., regular expression matching) and statistical language modeling to machine learning methods. Rule-based classification [5], used in the early work on QC, can successfully classify some questions, but fails to achieve satisfactory performance in general. Moreover, the classification rules must be compiled manually, making it difficult to adapt the classifier to new taxonomies.

More recent approaches to QC are machine learning-based. In [19] several machine learning methods were compared. The best classification results were achieved using SVM (80.2% accuracy), followed by decision trees (77% accuracy). SVM was also used in [4] and [11]. SNOW (Sparse network of winnows) learning architecture has also shown good performance (74.0% accuracy) [6]. As regards the features used for classification, they range from simple features such as words and ngrams [19], over syntactic features such as noun phrases, chunks, and head chunks [6,11], towards more semantic features such as named entities [4,6] and WordNet hypernyms [11].

As an alternative to machine learning methods for QC, statistical language modeling was also used [10,9]. Best results were achieved using two smoothing methods: improved absolute discounting and log-linear interpolation. Using these methods classification accuracy reached up to 80%, which is comparable to the results achieved using SVM.

The QC approaches also differ in what classification taxonomy they use. Early approaches used mainly one-level taxonomy consisting of a few coarse-grained classes, but later it was concluded that this is not enough to achieve satisfactory performance for real-world applications. In [6], a multilevel taxonomy was proposed, which was frequently used in much of subsequent work, e.g. [9,10,19]. The proposed taxonomy consists of two levels: a coarse-grained level that groups questions into six basic classes, and a fine-grained level consisting of 50 classes.

## 3 Question Classification for the Croatian Language

Considering the selected taxonomy, which consists of two levels (six coarse- and 50 fine-grained classes), the QC problem can be solved in basically three ways [6]: coarse-grained classification, fine-grained classification, and hierarchical fine-grained classification. In case of the latter, classification is performed in two steps: a question is first classified in one of the coarse-grained classes and then into one of the subordinated fine-grained classes. We use three classification algorithms: support vector machine (SVM), decision trees (DT), and k-nearest neighbors (k-NN), as well as language modeling (LM). It has been shown for English that SVM outperforms other methods on the QC problem, but we test other classifiers as well to verify whether the same holds for Croatian. For SVM we use SVMLib [2], for DT we use RapidMiner,[1] whereas for k-NN and LM we use our own implementation with Witten-Bell smoothing [16]. The language model is not a classification method per se, but it can be used for classification as follows. For each class we build a language model using questions from that class. We then use the models to estimate the probability of a sequence of words that make up a question, and classify the question into the class for which the corresponding model yields the highest probability. Because the taxonomy proposed in [6] is well accepted and widely applied, we also use this taxonomy here, allowing for easier comparison to the work of others.

### 3.1 Features

Classifier input are questions represented as $n$-dimensional feature vectors. Unlike [6,4], which use more complex features, we only use two basic features: words and bigrams. Prior to feature extraction, words from the question are lemmatized (various inflectional forms of a single word are conflated to a single representative form) using a morphological lexicon [14]. This reduces the number of features and potentially increases the classifier performance.

Bigrams are important features because they can, to a certain extent, capture the syntactic relations as reflected by the word order. Besides bigrams, we also use skip-bigrams, i.e., sequences of two words with one intervening word.

To reduce the dimensionality of the feature space (order of magnitude $10^4$), we filtered out words and bigrams whose frequency is less than two. Additionally, we experimented with three feature selection methods [18]: information gain (IG), $\chi^2$- statistic (CHI), and document frequency (DF) .

### 3.2 QC Test Collection

Papers dealing with QC for English language commonly use a test collection containing 5500 questions developed in [6].[2] Because there is no previous work on QC for the

[1] http://rapid-i.com/

[2] This collection was created as part of work described in [6] and is available at http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/

Croatian, we have built our test collection from scratch. The test collection consists of two groups of questions. The first group of questions (collection C1) was created by manually translating 1350 questions from [6], which were already classified according to taxonomy proposed by same authors.[3] The second group (collection C2) are 953 questions taken from the game show *"Who Wants to Be a Millionaire?"*.[4] These questions were not classified, so we manually classified them based on the same taxonomy. Collection C3 is the union of collections C1 and C2 and totals 2303 questions. The distribution of questions into classes is given in Table 1.

**Table 1.** The distribution of questions into classes for the three collections (%)

| Class | C1 | C2 | C3 | Class | C1 | C2 | C3 |
|---|---|---|---|---|---|---|---|
| ABBREVIATION | 1.56 | 1.26 | 1.43 | Term | 1.93 | 14.30 | 7.04 |
| Abbreviation | 0.22 | 0.53 | 0.35 | Vehicle | 0.30 | 0.00 | 0.17 |
| Expansion | 1.33 | 0.74 | 1.09 | Word | 0.37 | 0.00 | 0.22 |
| DESCRIPTION | 20.44 | 4.31 | 13.78 | HUMAN | 21.41 | 22.71 | 21.95 |
| Definition | 7.63 | 1.16 | 4.95 | Description | 1.04 | 0.00 | 0.61 |
| Description | 4.59 | 2.94 | 3.91 | Group | 3.04 | 1.47 | 2.39 |
| Manner | 4.59 | 0.00 | 2.69 | Individual | 16.67 | 21.24 | 18.56 |
| Reason | 3.63 | 0.21 | 2.22 | Title | 0.67 | 0.00 | 0.39 |
| ENTITY | 23.85 | 34.07 | 28.07 | LOCATION | 17.04 | 21.03 | 18.69 |
| Animal | 2.74 | 2.42 | 2.61 | City | 2.89 | 8.10 | 5.04 |
| Body | 0.37 | 0.00 | 0.22 | Country | 2.89 | 6.83 | 4.52 |
| Color | 0.67 | 1.37 | 0.96 | Mountain | 0.74 | 0.21 | 0.52 |
| Creative | 4.00 | 0.11 | 2.39 | Other | 9.19 | 5.05 | 7.48 |
| Currency | 0.00 | 0.00 | 0.00 | State | 1.33 | 0.84 | 1.13 |
| Dis.med. | 1.78 | 0.42 | 1.22 | NUMERIC | 15.70 | 16.61 | 16.08 |
| Event | 1.33 | 0.11 | 0.83 | Code | 0.22 | 0.84 | 0.48 |
| Food | 1.70 | 0.11 | 1.04 | Count | 6.89 | 7.68 | 7.21 |
| Instrument | 0.37 | 0.21 | 0.30 | Date | 4.15 | 5.15 | 4.56 |
| Lang | 0.37 | 0.74 | 0.52 | Distance | 0.44 | 0.21 | 0.35 |
| Letter | 0.30 | 0.74 | 0.48 | Money | 1.11 | 0.11 | 0.70 |
| Other | 3.26 | 11.78 | 6.78 | Order | 0.15 | 0.53 | 0.30 |
| Plant | 0.37 | 0.21 | 0.30 | Other | 0.22 | 0.84 | 0.48 |
| Product | 0.89 | 0.11 | 0.56 | Percent | 0.67 | 0.53 | 0.61 |
| Religion | 0.00 | 0.00 | 0.00 | Period | 0.96 | 0.32 | 0.70 |
| Sport | 0.96 | 0.42 | 0.74 | Size | 0.22 | 0.00 | 0.13 |
| Substance | 1.11 | 0.53 | 0.87 | Speed | 0.52 | 0.21 | 0.39 |
| Symbol | 0.22 | 0.53 | 0.35 | Temp | 0.07 | 0.00 | 0.04 |
| Technique | 0.81 | 0.00 | 0.48 | Weight | 0.07 | 0.21 | 0.13 |

[3] Collection C1 is available at `http://ktlab.fer.hr/download/cro-qa-test-c1.txt`

[4] *"Who Wants to Be a Millionaire?"* is a television game show licensed by Sony Pictures Television International. The Croatian version was aired on national television from year 2002 until 2010.

## 4 Evaluation

Preliminary experiments with SVM classifier revealed that DF outperformed other feature selection methods, making it possible to remove 60% of features without deterioration in classifier performance, thus we used DF in all subsequent experiments. Given the fact that evaluation is conducted for three different question collections and in three ways for each (coarse-grained, fine-grained, and hierarchical fine-grained classification), we had to train nine models of each classifier. Because model selection with cross validation would be too time consuming in this case, we performed hyperparameter optimization manually on a small subset of the training data.

The evaluation results for four classification methods are shown in Table 2. The result are given in terms of classification accuracy (Acc), micro-averaged precision (P), recall (R), and the F1 measure. The SVM classifier achieved the best results on all three collections, as well as for all three classification approaches (coarse-grained, fine-grained, and hierarchical fine-grained classification). The results for coarse-grained classification are better than for fine-grained classification, which is expected due to the smaller number of classes. The hierarchical fine-grained classification does not improve classification results, which has also been confirmed in [6] for English language.

Classifier performance on collection C1 consistently outperforms classifier performance on collection C2. Although this could be explained by the fact that collection C1 is somewhat larger than collection C2, this does not seem plausible as classifier performance is even worse on collection C3, the union of the two collections. However, collection C2 has a higher proportion of questions from the ENTITY class (24% questions in C1 vs. 34.5% question in C2), on which classifier performance tends to be worse (cf. Table 3).

Note that there are substantial differences between accuracy and macro-averaged values. This is because many classes have little or no questions (e.g., 11 classes in C2

**Table 2.** Classification results on three test collections

| | | Coarse-grained (%) | | | | Fine-grained (%) | | | | H. fine-grained (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Classifier | Collection | Acc | P | R | F1 | Acc | P | R | F1 | Acc | P | R | F1 |
| SVM | C1 | 85.7 | 81.0 | 76.8 | 77.9 | 70.2 | 39.1 | 37.3 | 36.9 | 69.4 | 37.9 | 37.2 | 36.2 |
| | C2 | 75.9 | 64.5 | 62.3 | 62.8 | 69.2 | 22.7 | 22.1 | 21.8 | 66.5 | 22.1 | 21.9 | 21.4 |
| | C3 | 83.3 | 81.5 | 76.7 | 78.0 | 69.9 | 43.4 | 39.1 | 39.4 | 69.8 | 42.2 | 39.3 | 39.2 |
| DT | C1 | 75.6 | 73.7 | 70.8 | 71.6 | 62.8 | 45.7 | 38.9 | 39.4 | 56.2 | 28.9 | 28.0 | 27.2 |
| | C2 | 68.5 | 69.4 | 66.6 | 66.2 | 62.4 | 25.2 | 22.3 | 20.8 | 57.4 | 17.1 | 15.7 | 15.7 |
| | C3 | 77.1 | 66.4 | 66.2 | 66.2 | 65.6 | 44.0 | 34.1 | 35.3 | 61.5 | 34.4 | 29.5 | 29.6 |
| k-NN | C1 | 75.9 | 71.6 | 71.2 | 70.4 | 60.8 | 32.8 | 33.0 | 31.2 | 60.6 | 31.2 | 32.3 | 30.0 |
| | C2 | 70.9 | 60.9 | 58.4 | 58.6 | 60.5 | 19.2 | 20.0 | 19.0 | 60.3 | 17.6 | 18.5 | 17.3 |
| | C3 | 74.6 | 73.3 | 72.2 | 71.9 | 60.7 | 35.5 | 35.9 | 34.0 | 60.8 | 35.6 | 35.6 | 33.7 |
| LM | C1 | 66.6 | 63.1 | 62.1 | 60.3 | 55.5 | 31.9 | 29.7 | 29.0 | 53.7 | 28.8 | 27.1 | 26.3 |
| | C2 | 60.9 | 56.0 | 52.9 | 52.4 | 53.0 | 18.3 | 17.6 | 17.2 | 50.6 | 17.9 | 17.2 | 16.8 |
| | C3 | 60.5 | 57.4 | 57.9 | 54.9 | 52.4 | 33.9 | 31.3 | 30.7 | 47.4 | 30.8 | 29.1 | 27.9 |

**Table 3.** Results of SVM coarse-grained classification on C1

| | Abbreviation | Entity | Description | Human | Location | Numeric |
|---|---|---|---|---|---|---|
| P (%) | 100.0 | 75.5 | 85.7 | 89.7 | 92.7 | 95.1 |
| R (%) | 38.1 | 85.4 | 88.0 | 84.1 | 83.0 | 92.9 |
| F1 (%) | 55.2 | 79.0 | 86.8 | 86.8 | 87.6 | 94.0 |

**Table 4.** Classification results on C3 using SVM with word forms, stems, and lemmas

| | Coarse-grained (%) | | | | Fine-grained (%) | | | | H. fine-grained (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | P | R | F1 | Acc | P | R | F1 | Acc | P | R | F1 |
| word forms | 82.3 | 79.6 | 75.8 | 76.5 | 67.7 | 41.5 | 36.4 | 37.0 | 67.7 | 41.3 | 37.1 | 37.5 |
| stems | 82.9 | 82.9 | 77.7 | 79.2 | 70.0 | 41.1 | 39.7 | 40.1 | 69.1 | 44.4 | 40.3 | 40.6 |
| lemmas | 83.3 | 81.5 | 76.7 | 78.0 | 69.9 | 43.4 | 39.1 | 39.4 | 69.8 | 42.2 | 39.3 | 39.2 |

**Table 5.** Classification results on the English counterpart of C1

| | Coarse-grained (%) | | | | Fine-grained (%) | | | | H. fine-grained (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Classifier | Acc | P | R | F1 | Acc | P | R | F1 | Acc | P | R | F1 |
| SVM | 78.7 | 78.8 | 75.0 | 76.1 | 64.2 | 32.4 | 30.2 | 29.9 | 64.7 | 33.3 | 31.4 | 30.1 |
| DT | 70.0 | 78.5 | 65.7 | 69.1 | 58.3 | 37.75 | 31.65 | 32.8 | 57.9 | 34.3 | 25.9 | 27.4 |
| k-NN | 70.7 | 70.1 | 68.2 | 68.1 | 57.1 | 29.4 | 30.1 | 27.9 | 55.5 | 28.7 | 28.8 | 27.0 |
| LM | 63.1 | 58.7 | 64.6 | 58.0 | 52.2 | 29.6 | 27.9 | 27.3 | 50.0 | 28.9 | 27.5 | 26.6 |

have no questions). Classifier performance on such classes is very low or equals zero, respectively, which affects the overall performance.

Table 3 shows the results of SVM coarse-grained classification on collection C1. Classifier performs best on the Numeric question type, and worst on Entity and Abbreviation types (the latter is a weakly represented class accounting for less than 2% of questions). As concerns the fine-grained classification into well-represented classes (30 or more questions per class), results are best for Location-City (F1=92.5%) question type and worst for Human-Group type (F1=37.5%).

We also experimented with using word forms and stems as features. The results for SVM on collection C3 are given in Table 4. Using word forms yields slightly lower results than using lemmas, whereas using stems yields similar results to using lemmas. Thus, it is better to eliminate the morphological variation altogether than to retain the morphosyntactic information conveyed by word forms. The same was also confirmed in [8] for topical classification.

To compare the differences in QC performance between Croatian and English, we trained and tested SVM on the English counterpart of collection C1. As shown in Table 5, the classifier performance on English data is lower for all classifiers except for LM. This is contrary to the results for topical classification reported in [8]. We hypothesize that this is due to lexical differences between Croatian and English (Croatian words might be more discriminative for QC), but this is certainly something worth further investigation.

## 5 Conclusion

The performance of a QA system depends on its ability to correctly classify a question according to the answer type. The aim of this paper was to study machine learning methods for question classification for the Croatian language. We experimented with four methods (SVM, k-NN, decision trees, and language model). As part of this work we have create a Croatian QC test collection consisting based on a two-level question taxonomy. We experimented with fine-grained, coarse-grained, and hierarchical fine-grained classification. The SVM outperformed other classifiers, and – as expected – the results are better for coarse-grained classification. However, hierarchical fine-grained classification did not perform better than direct fine-grained classification. Results are slightly better when using morphological normalization (stemming or lemmatization), and – somewhat surprisingly – better on Croatian than on English data.

Further work includes the improvement of the QC test collection. It is necessary to supplement the collection with more questions, in particular the addition of questions from weakly represented classes. We also plan to incorporate more semantic features than words and bigrams, such as named entities and synonyms.

**Acknowledgments.** This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under the Grant 036-1300646-1986. We thank the anonymous reviewers for their useful comments.

## References

1. Čeh, I., Ojsteršek, M.: Developing a question answering system for the Slovene language. WSEAS Transactions on Information Science and Applications 6(9), 1533–1543 (2009)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) software, http://www.csie.ntu.edu.tw/~cjlin/libsvm
3. Green, B.F., Wolf, A.K., Chomsky, C., Laughery, K.: Baseball: An automatic question answerer. In: Proceedings of the Western Joint Computer Conference, vol. 19, pp. 219–224 (1961); reprinted in Grosz et al (1986)
4. Hacioglu, K., Ward, W.: Question classification with support vector machines and error correcting codes. In: NAACL 2003: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 28–30 (2003)
5. Kwok, C., Etzioni, O., Weld, D.S.: Scaling question answering to the web. ACM Trans. Inf. Syst. 19(3), 242–262 (2001)
6. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th COLING, pp. 556–562. Association for Computational Linguistics (2002)
7. Magnini, B., Vallin, A., Ayache, C., Erbach, G., Peñas, A., De Rijke, M., Rocha, P., Simov, K., Sutcliffe, R.: Overview of the CLEF 2004 multilingual question answering track. Multilingual Information Access for Text, Speech and Images, 371–391 (2005)

8. Malenica, M., Šmuc, T., Šnajder, J., Dalbelo Bašić, B.: Language morphology offset: Text classification on a Croatian-English parallel corpus. Information Processing and Management 41(1), 325–339 (2008)
9. Merkel, A., Klakow, D.: Improved methods for language model based question classification. In: Proceedings of 8th Interspeech Conference, Antwerp (2007)
10. Merkel, A., Klakow, D.: Language model based query classification. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 720–723. Springer, Heidelberg (2007)
11. Metzler, D., Croft, W.B.: Analysis of statistical question classification for fact-based questions. Journal of Information Retrieval 8, 481–504 (2004)
12. Simov, K., Osenova, P.: BulQA: Bulgarian-Bulgarian question answering at CLEF 2005. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 517–526. Springer, Heidelberg (2006)
13. Voorhees, E.: The TREC question answering track. Natural Language Engineering 7(04), 361–378 (2001)
14. Šnajder, J., Dalbelo Bašić, B., Tadić, M.: Automatic acquisition of inflectional lexica for morphological normalisation. Information Processing and Management 44(5), 1720–1731 (2008)
15. Walas, M., Jassem, K.: Named entity recognition in a Polish question answering system. Information Retrieval, 1–10 (2003)
16. Witten, I.H., Bell, T.C.: The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. IEEE Transactions on Information Theory 37(4), 1085–1094 (1991)
17. Woods, W.A.: Progress in natural language understanding: an application to lunar geology. In: Proceedings of American Federation of Information Processing Societies Conference, AFIPS 1973, pp. 441–450. ACM, New York (1973)
18. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of ICML 1997, 14th International Conference on Machine Learning, Nashville, US, pp. 412–420 (1997)
19. Zhang, D., Lee, W.S.: Question classification using support vector machines. In: SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 26–32. ACM, New York (2003)

# Random Indexing Distributional Semantic Models for Croatian Language

Vedrana Janković, Jan Šnajder, and Bojana Dalbelo Bašić

Faculty of Electrical Engineering and Computing,
University of Zagreb, Croatia
{vedrana.jankovic,jan.snajder,bojana.dalbelo}@fer.hr

**Abstract.** Distributional semantic models (DSMs) model semantic relations between expressions by comparing the contexts in which these expressions occur. This paper presents an extensive evaluation of distributional semantic models for Croatian language. We focus on random indexing models, an efficient and scalable approach to building DSMs. We build a number of models with different parameters (dimension, context type, and similarity measure) and compare them against human semantic similarity judgments. Our results indicate that even low-dimensional random indexing models may outperform the raw frequency models, and that the choice of the similarity measure is most important.

**Keywords:** Distributional semantic model, computational semantics, random indexing, Croatian language.

## 1 Introduction

Automating the construction of semantic representations of natural language expressions requires as its prerequisite the definition of meaning. This definition is one of the central linguistic and philosophical problems and has been approached from, among others, conceptual, objectified, semiotic, and pragmatic perspective [24]. A purely pragmatic approach manifests itself in distributional semantics, in which the meaning is defined as a relative, and not absolute concept, relying on *distributional hypothesis*: words similar in meaning occur in similar contexts, i.e., distributional similarity can be used to estimate meaning similarity because of the existing correlation between these two [20]. Distributional semantic models (DSMs) represent meanings of lexical expressions and their semantic relations as multi-dimensional vectors, capturing the context in which the modeled expressions occur in the corpora [11]. Another approach based on the distributional hypothesis is *word sketches* – automatic, corpus-based summaries of a word's grammatical and collocational behaviour [9]. Distributional semantic models have a wide range of possible applications in lexicography, natural language processing, and information retrieval.

The focus of this paper are DSMs for Croatian language. We perform an extensive evaluation of DSMs built by varying three main model parameters: the dimension, context type, and similarity measure. In order to determine which model works best for Croatian language, we evaluate the models against human semantic similarity judgments. We focus in particular on the random indexing DSMs [18] because of their computational efficiency, and compare them against the raw frequency models. The rest of

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 411–418, 2011.

the paper is organized as follows. Section 2 formally defines a DSM and shortly reviews the related work. Section 3 describes our DSMs for Croatian language, while Section 4 presents evaluation and results. Section 5 concludes the paper and outlines future work.

## 2 Distributional Semantic Models

A distributional semantic model is defined as a co-occurrence matrix $M$, where each row represents a context vector, i.e., the distribution of a target element across contexts [7]. DSM can be formally represented as a tuple:

$$DSM = (T, C, R, W, M, d, S),$$

where $T$ are target elements (e.g., words) whose meaning is being modeled, $C$ is the context in which target elements are observed, $R$ is the relation between target elements and the context, $W$ is the context weighting scheme, $M$ is a distributional matrix $|T| \times |C|$, function $d : M \rightarrow M'$ is the dimension reduction function, and $S$ is the distance measure between vectors from $M$ or $M'$. The most basic DSM is a raw frequency model, where values of $M$ are the raw frequencies of the context elements. Numerous other DSMs have been proposed, including LSA (Latent Semantic Analysis) [10], HAL (Hyperspace Analogue to Language) [5], Dependency-Based Semantic Space Models [16], Distributional Memory [1], and Random Indexing, also known as Random Projection [8,18,2]. A more detailed overview can be found in [6,19,22].

DSMs have been applied to a number of Slavic languages, including Bulgarian [14], Czech [21], Polish [17,3,4], and Russian [15,13]. To the best of our knowledge, no previous DSM research has been done for Croatian language except for [12], which compares eight raw frequency DSMs, differing in measures of semantic similarity.

The focus of this paper are random indexing DSMs. Random indexing (RI) is a dimensionality reduction technique in which a random matrix $M_R$ of dimensions $|C| \times k$ is used to project the original matrix $M$ of dimensions $|T| \times |C|$ to a reduced matrix $M'$ of dimensions $|T| \times k$, $k \ll |C|$. The $k$-dimensional rows of $M_R$ are called index vectors. Each index vector $\mathbf{r}_j$ represents one context element $c_j \in C$. The index vectors are sparse vectors with a small number $\eta$ of randomly distributed $+1$ and $-1$ values, the remaining values being $0$. Given a target element $t_i$, its context vector $\mathbf{t}_i$ is constructed by considering all context elements $c_j$ occurring within the context of $t_i$, i.e., the elements for which $R(t_i, c_j)$ holds. For every $c_j$, the corresponding random index vector $\mathbf{r}_j$ is added to vector $\mathbf{t}_i$. The key idea behind RI is that the distance between points in the randomly selected subspace of a high enough dimension $k$ is approximately preserved [2]. Thus, RI performs an implicit dimensionality reduction, making RI computationally more efficient and scalable than other DSM approaches.

## 3 Building DSMs for Croatian Language

### 3.1 Corpus

The corpus used in our experiments is a collection of articles from the Croatian newspaper Vjesnik from year 1999 to 2009.[1] The corpus originally consists of 276 231

[1] www.vjesnik.hr

documents and 85M tokens. We have preprocessed the corpus by case folding and by removal of punctuation, special characters, digit tokens, and stop-words. Words shorter than three characters have also been removed and the corpus has been lemmatized [23], after which hapax and dis legomena have been filtered out. We use an ambiguous lemmatization procedure, which retains all candidate lemmas of a given word form. The preprocessed corpus consists of 49M tokens and 155 274M types.

### 3.2 Models

As the target elements, we have selected 185 most frequent words from the corpus. In total, we have built 350 models: 70 raw frequency models and 280 RI models. The models differ in context type and similarity measure used, while RI models also differ in random index vector dimension $k$.

The context can be (1) a sentence, (2) a symmetric, or (3) an asymmetric window, and may be either weighted or unweighted. Symmetric and asymmetric windows span either 5, 10, and 20 words around the target word (symmetric window) or words on the left or the right of the target word (asymmetric window), regardless of sentence boundaries. As a context weighting scheme we used the double L-function:

$$ll(x,\alpha,\beta,\gamma,\delta) = \begin{cases} l(x,\alpha,\beta), & x \geq 0 \\ l(-x,\gamma,\delta), & x < 0 \end{cases}, \quad l(x,\alpha,\beta) = \begin{cases} 1 & x < \alpha \\ \frac{\beta - x}{\beta - \alpha} & \alpha \leq x \leq \beta \\ 0 & \beta < x \end{cases} \tag{1}$$

For example, a symmetric weighted window spanning five words is obtained with $ll(x,1,5,1,5)$. For asymmetric windows we used no weighting schemes. For index vector dimension $k$ we used 100, 200, 500, and 1000, with $\eta$ = 2, 4, 6, and 8, respectively. We have calculated the similarity between $\binom{185}{2} = 17,020$ target word pairs (each represented by a context vector pair) for all 350 models, using each of the following five similarity measures: Manhattan distance, Euclidean distance, Cosine distance, Dice coefficient, and Jaccard coefficient. Because the latter two are defined only for binary vectors, we have used the modified formulas proposed by [6]. All pairwise similarity grades have then been scaled to a $[1,5]$ interval, 5 denoting the highest vector similarity.

## 4 Evaluation and Results

### 4.1 Semantic Similarity Judgments

Twelve judges were given 450 word pairs manually selected from 17,020 possible target word pairs and instructed to estimate the strength of semantic relationship on a scale from 1 to 5, with 5 being the strongest relation. The notion of semantic relation has been defined as a consolidation of both paradigmatic semantic relations (antonymy, hyponymy, cohyponymy, meronymy, and synonymy) and syntagmatic relations that capture lexicalized expressions (such as idioms, compound nouns, and clichés), as well as derivational relations and broader associative relations. For polysemous words, the judges were instructed to choose the reading that maximizes the relation strength of the

**Table 1.** Examples of human similarity judgments

| | | Average grade | |
|---|---|---|---|
| Word pair | | 6 judges | 12 judges |
| *politika – politički* | (*politics – political*) | 5.0 ± 0.0 | 4.9 ± 0.3 |
| *igra – igrač* | (*game – player*) | 5.0 ± 0.0 | 4.8 ± 0.4 |
| *kuna – novac* | (*kuna – money*) | 5.0 ± 0.0 | 4.8 ± 0.4 |
| *početak – kraj* | (*beginning – end*) | 5.0 ± 0.0 | 4.7 ± 0.5 |
| *zemlja – država* | (*country – state*) | 5.0 ± 0.0 | 4.6 ± 0.8 |
| *reći – govoriti* | (*say – speak*) | 4.8 ± 0.4 | 4.8 ± 0.4 |
| *imati – nemati* | (*have – not have*) | 4.8 ± 0.4 | 4.7 ± 0.5 |
| *banka – novac* | (*bank – money*) | 4.8 ± 0.4 | 4.5 ± 0.5 |
| *utorak – tjedan* | (*Tuesday – week*) | 4.8 ± 0.4 | 4.2 ± 1.2 |
| *dolar – novac* | (*dollar – money*) | 4.7 ± 0.5 | 4.7 ± 0.5 |
| *utorak – tjedan* | (*Tuesday – week*) | 4.8 ± 0.4 | 4.2 ± 1.2 |
| *kuna – dolar* | (*kuna – dollar*) | 4.7 ± 0.5 | 4.3 ± 0.8 |
| *član – klub* | (*member – club*) | 4.7 ± 0.5 | 4.2 ± 0.7 |
| *djeca – škola* | (*children – school*) | 4.7 ± 0.5 | 4.1 ± 0.8 |
| *utorak – srijeda* | (*Tuesday – Wednesday*) | 4.5 ± 0.8 | 4.1 ± 0.9 |
| *sat – tjedan* | (*hour – week*) | 4.5 ± 0.6 | 4.1 ± 0.7 |
| *njemački – francuski* | (*German – French*) | 4.5 ± 0.6 | 4.0 ± 0.9 |
| *pravi – trenutak* | (*right – moment*) | 4.5 ± 0.6 | 3.9 ± 0.8 |
| *vojni – snaga* | (*military – power*) | 3.8 ± 0.8 | 3.9 ± 0.8 |
| *sustav – izjaviti* | (*system – declare*) | 1.7 ± 1.2 | 1.5 ± 0.9 |

word pair. In this way we avoid the inconsistencies arising from different treatments of polysemous words.

Fleiss' kappa was used to calculate pairwise agreement and total inter-judge agreement. As expected with semantic annotation tasks, the agreement is rather low: $\kappa = 0.27$ (fair agreement). To improve the agreement, we decided to select a subset of judges with stronger mutual agreement. To this end, we generated a $12 \times 12$ pairwise agreement matrix, which we then clustered using hierarchical agglomerative clustering with average linkage. From this, a group of 6 judges with stronger mutual agreement was formed ($\kappa = 0.35$; fair agreement). From these two groups we have created two averaged similarity vectors (gold standard vectors) for further evaluation. Table 1 shows exemplary word pairs in different semantic relations. For each word pair, average grade and standard deviation for both judge groups is given. As expected, the standard deviations in grades for 6 annotators are in principle less than those for 12 annotators.

### 4.2 Results

All 350 model-generated similarity vectors were compared against the two gold standard vectors using Mean Square Error (MSE). Selected results are given in Table 2, ranked in increasing order of MSE. Model type is raw frequency (Raw) or random indexing (RI-$k$), where $k$ is the index vector dimension. Context is either sentence (S),

**Table 2.** Mean Square Error for the selected DSMs

| Model | | | Mean Square Error (Rank) | | | |
|---|---|---|---|---|---|---|
| Type | Context | Similarity | 6 judges | | 12 judges | |
| RI–100 | W–0L–5R | Dice | 1.96 | (1) | 1.68 | (5) |
| RI–100 | W–0L–10R | Dice | 1.98 | (3) | 1.64 | (1) |
| RI–500 | S | Dice | 1.98 | (4) | 1.69 | (6) |
| RI–200 | Ww–5L–5R | Dice | 2.00 | (5) | 1.65 | (3) |
| RI–1000 | W–20L–0R | Dice | 2.09 | (9) | 1.95 | (18) |
| RI–200 | W–10L–10R | Dice | 2.10 | (11) | 1.70 | (7) |
| RI–1000 | W–0L–5R | Jaccard | 2.13 | (13) | 1.96 | (22) |
| RI–500 | Sw | Jaccard | 2.24 | (25) | 2.03 | (30) |
| Raw | Ww–5L–5R | Jaccard | 2.27 | (31) | 2.15 | (42) |
| RI–200 | Ww–5L–5R | Cosine | 2.60 | (59) | 2.72 | (67) |
| RI–200 | Ww–5L–5R | Euclidean | 5.97 | (186) | 6.37 | (186) |
| RI–200 | Ww–5L–5R | Manhattan | 6.23 | (190) | 6.66 | (192) |
| RI–1000 | W–20L–20R | Dice | 7.94 | (331) | 8.60 | (331) |
| RI–100 | W–0L–20R | Dice | 8.46 | (350) | 9.16 | (350) |

weighted sentence (Sw), window (W), or weighted window (Ww). Left and right window spans, $x$ and $y$, respectively, are given by $x$L–$y$R. Results include the best performing model for each similarity measure and the best performing raw frequency model.

The results indicate that RI models outperform raw frequency models (best raw frequency model is ranked 31 and 42 compared to the 6 and 12 judges, respectively). In particular, for all considered dimensions $k$, there exists a RI model that outperforms all raw frequency models. Surprisingly, the results also suggest that the performance of an RI model does not necessarily improve with increases in dimension. It is evident that the choice of the similarity measure is an important one: the best performing models use the Dice or Jaccard coefficient, whereas models using Manhattan or Euclidean distance perform poorly. Interestingly, there seems to be no correlation between context type used and model performance. The best performing models differ greatly in context types, while, on the other hand, models with similar context types greatly differ in performance (e.g., models ranked 3 and 350 compared to the 6 judges). Thus, the results are inconclusive with regard to how weighting, window symmetry, and span affect the performance. Also, no consistent difference has been observed between performances evaluated against the two gold standard vectors.

It is interesting to compare human similarity judgments against the similarity scores generated by the best performing DSM. Although, as noted in [17], the analysis of model-generated similarity scores of arbitrary chosen word pairs may be misleading, we still can identify some general trends. Cohyponymes and hyponymes seem to score high on the model-generated list (11 out of top 25 ranked word pairs), while the rest are syntagmatically related words. Contrary to this, human similarity judgments are highest for synonyms, antonyms, and derivationally related words.

### 4.3 Remarks

Several factors have to be considered when interpreting the results. Firstly, DSMs blur different readings of a polysemous word into a single vector, whereas human judges select one particular reading. This introduces an error when comparing similarity grades for ambiguous word pairs. Secondly, DSMs are corpus-specific and reflect the semantic relations as found in the corpus. Because our DSMs are built on a domain-specific corpus, they are not as representative of a language as they would be if they were build on a general language corpus. On the other hand, it is exactly because our corpus is domain-specific that we expect the degree of polysemy to be lower than that for a general language corpus.

In this paper we evaluated the DSMs on word pairs comprised of 185 most frequent words from the corpus. As noted by one reviewer, it is much easier to build semantic models for frequent than for infrequent words. With frequent words, however, we get more reliable statistical estimates and more conclusive results. It can also be argued that DSMs that perform bad on frequent words are not likely to perform well on the less frequent words. In either case, an analysis of DSM performance with respect to word frequencies deserves a more thorough investigation, which we leave for future work.

Perhaps the biggest limitation of our evaluation are the low inter-judge agreement scores, which make the results less reliable. We have tried to alleviate this drawback by analyzing the inter-judge agreement and selecting the judges with the highest mutual agreement. It can be argued – as one of the reviewers has – that quantitative evaluation of semantic similarities is very hard or even impossible for humans. Nevertheless, we believe that evaluation based on human similarity judgments is useful. As shown in Table 2, there are significant differences in MSE values between different models, and these results do tell us which models are better and which are worse. More reliable human similarity judgments, and thus more conclusive results, could perhaps be obtained with a larger number of judges.

## 5 Conclusion

Distributional semantic models (DSMs) model semantic relations between expressions by comparing the contexts in which they occur. Random indexing is an efficient technique for building DSMs, which performs an implicit dimensionality reduction. In this paper we evaluated 350 DSMs for Croatian language, with a focus on RI models. The models were evaluated by comparison against human gold standard judgments. The results suggest that the choice of the similarity measure is an important one, while the influence of other model parameters is less clear. As expected, random indexing models outperform the raw frequency models; this is the case even with low-dimensional models.

As part of future work we intend to experiment with different corpora and weighting schemes. We also intend to experiment with other DSMs derived from the raw frequency model and compare these against RI models. A more detailed analysis with respect to semantic relation types could provide us with insights whether some models are better at capturing certain relation types.

**Acknowledgments.** This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under the Grant 036-1300646-1986. We thank the anonymous reviewers for their useful comments.

## References

1. Baroni, M., Lenci, A.: One distributional memory, many semantic spaces. In: Proceedings of the EACL Workshop on Geometrical Models of Natural Language Semantics (2009)
2. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: KDD 2001: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2001)
3. Broda, B., Derwojedowa, M., Piasecki, M., Szpakowicz, S.: Corpus-based semantic relatedness for the construction of polish wordnet. In: Proceedings of the Sixth International Language Resources and Evaluation, LREC 2008 (2008)
4. Broda, B., Piasecki, M.: Supermatrix: a general tool for lexical semantic knowledge acquisition. In: Speech and Language Technology, vol. 11, pp. 239–254. Polish Phonetics Assocation (2008)
5. Burgess, C., Lund, K.: Modelling parsing constraints with high-dimensional context space. Language and Cognitive Processes 12, 1–34 (1997)
6. Curran, J.: From Distributional to Semantic Similarity. Ph.D. thesis, University of Edinburgh (2008)
7. Evert, S., Lenci, A.: Foundations of distributional semantic models, http://wordspace.collocations.de/lib/exe/fetch.php/course:acl2010:naacl2010_part1.slides.pdf (2010)
8. Kanerva, P.: Sparse Distributed Memory. MIT Press, Cambridge (1988)
9. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The sketch engine. In: Proceedings of the 11th EURALEX International Congress, pp. 105–116 (2004)
10. Landauer, T., Dumais, S.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review 104(2), 211–240 (1997)
11. Lenci, A.: Distributional semantics in linguistic and cognitive research. Italian Journal of Linguistics 20(1), 1–31 (2008)
12. Ljubešić, N., Boras, D., Bakarić, N., Njavro, J.: Comparing measures of semantic similarity. In: Proceedings of the ITI 2008 30th International Conference of Information Technology Interfaces (2008)
13. Mitrofanova, O., Mukhin, A., Panicheva, P., Savitsky, V.: Automatic word clustering in Russian texts. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 85–91. Springer, Heidelberg (2007)
14. Nakov, P.: Latent semantic analysis for bulgarian literature. In: Proceedings of Spring Conference of Bulgarian Mathematicians Union. Borovetz (2001)
15. Nakov, P.: Latent semantic analysis for russian literature investigation. In: Proceedings of the 120 years Bulgarian Naval Academy Conference, Citeseer (2001)
16. Pado, S., Lapata, M.: Dependency-based construction of semantic space models. Computational Linguistics 33(2), 161–199 (2007)
17. Piasecki, M.: Automated extraction of lexical meanings from corpus: A case study of potentialities and limitations. In: Representing Semantics in Digital Lexicography. Innovative Solutions for Lexical Entry Content in Slavic Lexicography, pp. 32–43. Institute of Slavic Studies, Polish Academy of Sciences (2009)

18. Sahlgren, M.: An introduction to random indexing. In: Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (2005)
19. Sahlgren, M.: The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. Ph.D. thesis, Department of Linguistics, Stockholm University (2006)
20. Sahlgren, M.: The distributional hypothesis. Rivista di Linguistica 20(1) (2008)
21. Smrž, P., Rychlỳ, P.: Finding semantically related words in large corpora. In: Matoušek, V., Mautner, P., Mouček, R., Tauser, K. (eds.) TSD 2001. LNCS (LNAI), vol. 2166, pp. 108–115. Springer, Heidelberg (2001)
22. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37, 141–188 (2010)
23. Šnajder, J., Dalbelo Bašić, B., Tadić, M.: Automatic acquisition of inflectional lexica for morphological normalisation. Information Processing and Management 44(5), 1720–1731 (2008)
24. Wilks, Y., Charniak, E.: Computational Semantics: An Introduction to Artificial Intelligence and Natural Language Understanding. North-Holland, Amsterdam (1976)

# Structure Annotation in the Polish Corpus of Suicide Notes

Michał Marcińczuk[1], Monika Zaśko-Zielińska[2], and Maciej Piasecki[1]

[1] Institute of Informatics, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, Wrocław, Poland
{michal.marcinczuk,maciej.piasecki}@pwr.wroc.pl
[2] Institute of Polish Philology, University of Wroclaw,
pl. Nankiera 15, Wrocław, Poland,
monik@uni.wroc.pl

**Abstract.** Polish Corpus of Suicide Notes (henceforth PCSN) is constructed to meet the needs of forensic linguistics. Suicide notes are messages created in borderline situation, shortly before death. Hence the annotation schema requires a complex description of a document structure, the textual content, as well as its linguistic properties. TEI was selected as the basis for the document encoding schema. TEI adaptation and extension with respect to such aspects of encoding as: a letter structure, various layers of changes and omissions, error correction, and extra-linguistic elements etc., are discussed with examples.

**Keywords:** forensic linguistics, suicide note, structure annotation.

## 1 Introduction

Forensic linguistics is a branch of applied linguistics and one of its important applications is preparing linguistic evidence for the court. An opinion of a linguist who appears in the court as an expert witness must be based on the objective linguistic data not on intuition. General corpora are used as a referential material for comparison [1], however construction of specialized corpora for particular types of forensic texts is the key issue for the forensic research [11]. Forensic corpora are typically small, include short texts and are used for e.g., authorship attribution and text authenticity analysis. Comparison within the same type of text is the key tool in the description of idiolect focused on separating text type features from the personal linguistic features of the speaker.

Our main objective is to develop an annotated corpus of suicide notes (the last messages written by a person before committing a suicide; the message can be in a form of letter, note, SMS or other), which is intended to become a reliable data source for linguistic analysis of new individual suicide notes. The analysis can be later applied in the court.

Application of computational methods to suicide notes started with the work of Shneidman [2], who investigated differences between genuine and simulated notes. Two sets of suicide notes (33 genuine and 33 simulated) were compared by using a system of categories which encompassed a set of predefined tags referring to roles, objects, emotional states, actions, institutions, statuses, qualities, symbolic referents [3]. These

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 419–426, 2011.

two sets of suicide notes were also used in [4]. Recently, Pestian et al. [5] applied machine learning methods to the recognition of false suicide notes. Their results are very positive, but their claimed superiority in comparison to the accuracy of humans causes doubts concerning the used evaluation method. "Vienna Corpus of Suicide Notes" was aimed at building psycholinguistics descriptions. It consists of suicide notes collected from years 2002–2005 and was used for comparing suicide note-writers with suicide non-note-writers. The comparison was based on eight variables: age, gender, marital status, occupation, psychiatric care, suicide motive and suicide method [6]. Additionally, analysis of suicide notes by forensic linguistic services is also a part of the general forensic document examination (e.g. ALIAS: software for forensic linguistic analysis[1]) performed with the help of the police corpora for crime investigation (e.g. suicide notes corpus from British Transport Police [7]).

A suicide note is on average a rather short piece of writing which is thematically and stylistically varied. Instead of performing a massive statistical analysis we have to strive for every piece of information characterising the text. Besides pure linguistic features, e.g. lexical or syntactical, pragmatic or even extra-linguistic features, e.g. the text structure and layout, can be also a valuable information source. Thus the annotation of a suicide note should encompass both layers: linguistic and structural. The latter is aimed specifically for forensic text analysis and support for tasks like suicide prediction. Structural information is mostly neglected in existing annotated corpora of suicide notes. Linguistic variables, like parts of speech and lexical frequency, as well as structure variables commonly used in quantitative text analysis, like the number of paragraphs and sentences, the length of sentences, etc., must be covered by the corpus annotation. However, we want to broaden this set. Only a subset of the functionality of the existing software for the forensic handwriting examination can be adapted to our task. For instance Wanda Workbench software supports annotation of content, material, script and writer but without structure annotation. The only elements of the structure that can be described in Wanda are characters and their selected measurable features. Text segmentation is still a significant problem in the forensic document pre-processing with OCR system [8].

The proposed suicide note structure annotation was inspired by three factors: handwritten form of the text, current forensic practice and requirements of the given text genre. We considered two types of text: suicide notes and Polish personal letters which are the most familiar type of an informally written text.

## 2 Choosing Text-Encoding Standard

Text-encoding format should facilitate the intended use not determine it. Thus, before deciding about the particular encoding standard to be selected, we have identified several aspects that should be covered by the annotation:

1. Text structure and layout:
   - formal letter structure: opening, body, closer,
   - physical text division into text blocks and lines, e.g., paragraphs, marginalia, page and line breaking etc.,

[1] http://www.aliastechnology.com

- text block layout, text alignment, indention, relative position,
- text formatting, e.g., bold, italic, underline, etc.,
- text omissions, deletion and insertion,

2. Correction
   - text correction introduced by authors and editors,
3. Linguistic information
   - segmentation into tokens and language expressions of various complexity,
   - semantic and pragmatic classification of text elements, e.g., salutation inside text, signature, envelope date expressed in different ways,
   - proper names,
4. Meta-data:
   - information about author,
   - physical description (paper format, type, etc.), linguistic description (type of text, e.g., letter, part of web log, statement).

In the above classification the most important seems to be the distinction between the description of the structure marked visually and linguistic properties of language expressions, that are independent from the former, e.g. salutation can occur as embedded inside a paragraph, not only in a separate text block.

We considered several standards for text representation, like TEI P5[2], XCES[3], KAF (*Kyoto Annotation Format*)[4] [9] and ISO TC 37/SC 4[5], as well as several formats developed for specific projects, like SCOTS or CEEC. Finally, TEI P5 was selected as it conforms to most our requirements and provides guidelines for manuscript description. TEI P5 also allows for the description of both the word-level and the medium of the document. It facilitates annotation of text segmentation, additions, ornaments, figures, underlining, crossing out, etc. TEI provides generic encoding guidelines for personal letters (among other genres) that can be further specified, e.g. with respect to cultural characteristics. This path was followed, e.g., in DALF[6], *Repertorium* project[7], *Vincent van Gogh - The Letters*[8], DBNL[9] and CARDS[10]. TEI allows to create the structure annotation which takes into account handwriting features, need for text reconstruction and writer identification. We follow this approach and use TEI as a basis for our corpus encoding. As not all of our requirements are met, e.g., handling hyphenated words, a further extension will be proposed.

[2] http://www.tei-c.org/Guidelines/P5/

[3] http://www.xces.org/

[4] http://xmlgroup.iit.cnr.it/kyoto/?option=com_content&view=article&id=141

[5] http://www.tc37sc4.org/

[6] http://www.kantl.be/ctb/project/dalf/

[7] http://clover.slavic.pitt.edu/repertorium/

[8] http://vangoghletters.org/vg/

[9] http://www.dbnl.org/

[10] http://alfclul.clul.ul.pt/cards-fly/index.php?page=mainen

## 3 Annotation Scheme

Annotation scheme is organised along several layers: physical layout (see Sec. 3), segmentation and morphological description[11], meta-data and semantic annotation.

Our starting point for encoding the physical and logical structure of the letter was DALF adaptation of TEI [10]. The letter body is divided into three parts: letter opener (**<opener>**), letter content (paragraph sequence **<p>**), letter closer (**<closer>**). Additionally, suicide notes are sometimes found in envelopes. More frequently a note includes the first page with information about recipient and the way of the suicide note delivery. Both are encoded by <envelope> tag: the envelop and the first page. In general the main parts are block elements and enclose complete lines of text. In some cases the opener and closer have a non-standard form, i.e. they overlap with a content paragraph, e.g., the first paragraph starts in the same line as the opener. To encode this **rend="inline"** attribute was used for opener and closer tags.

**<opener>** block can include three block elements: **<dateline>** – a date line occurring on the top of the letter, **<head>** – a title line and **<p>** – any other piece of text included in the opener. The last one can be repeated. **<closer>** can include three types of elements: **<dateline>** – in some letters the dateline appears not at the beginning but the end of the letter (but only once), **<p>** – a text passage (one and more) and postscript (**<ps>**). **<ps>** can encompass one **<p>** block element with two possible interpretations: **<p type="meta">** (*meta-paragraph*) to encode how the postscript section is introduced and **<p>** for a text block included. In some notes the meta-paragraph appears in the same line as the first postscript. Those cases are encoded by **rend="inline"** in the meta-paragraph. A postscript is both: a text block marked by the writer as PS, as well as, a text occurring after the signature. A postscript can be also introduced by a description, e.g. *I am adding*, *I'm also adding*, etc.

The line breaking **<lb/>** can appear in paragraphs (**<p>**) but also in date lines (**<dateline>**), when a date or a city name does not fit in one line, and in the title lines (**<head>**). Other elements consists of blocks, like **<opener>**. Page breaking inside **<p>** is expressed with **<pb/>** tag. It closes any open block element, i.e. divides one continuous text paragraph into two **<p>**. Specific visual separation of paragraphs (e.g. *a drawn line*) is described by <ornament/> tag, whose type attribute expresses the shape, e.g., *line*, *space*, *wave*, etc. (open list).

The horizontal alignment of text in block elements: **<p>**, **<head>**, **<dateline>** is stored in **rend** attribute with the following values: *left*, *centre*, *right*, *indent*, *margin-left*, *step-left*, *step-indent* and *step-center*. The **step-*** value describes the case in which the following line have bigger indention then the previous one. This is a characteristic feature of the Polish handwritten personal letters. The layout description of a block includes its positions as no location can be assumed as the default one.

Finally, pieces of text added in different places on the page (marginalia, doodles, etc.) are called *additions*. As the suicide notes layout has an atypical form (because of the context of writing e.g., lack of paper and emotional situation) we have to use variety means for its description: additions and paragraphs with the specified position which were written after the main text had been closed.

[11] Morphological description will be included in the final version of the corpus.

Not all parts of handwritten text can be read with enough confidence or are text in fact (e.g., drawings, signatures, etc.). All illegible fragments are annotated by **`<gap/>`** tag with four sub-types: *illegible* – a fragment impossible to read, a part is missing for some reason, *prosecutor* – a fragment obliterated by prosecutor (due to *anonymisation*) and *signature* – author's signature. If a text can be read but with some uncertainty, it is marked by **`<unclear>`** tag with a specified level of certainty (*low*, *medium* or *high*).

Any symbols or drawings that are not a sequence of characters are represented with **`<figure/>`** tag with type attribute describing the shape, e.g. *arrow*, *cross*, *emotikon*, *heart*, *other*, etc. (an open list).

For text replacement, deletion, addition etc. we use a combination of **`<del>`** and **`<add>`** tags. In contrast to **`<gap>`** the **`<del>`** tag is used when a piece of text is strikeout in some way but still readable. **`<add>`** (in contrast to **`<additions>`**) is used for a text inserted, e.g. between words or instead of strikeout word).

We distinguished two types of corrections made by: the author during writing (self-correction) or the editor during transcription. Editor corrections are important for automatic text processing. They are annotated by **`<corr>`** tag which encodes also the type of misspelling with several types predefined in the annotation guidelines. This description facilitates evaluation of writer's spelling competence.

Both correction types are distinguished by **`resp`** attribute with two values: *author* and *editor*. In the case of editorial corrections the original text is kept in **`sic`** attribute and the corrected text is put inside the tags.

The text formatting is described by **`<hi>`** tag with rend attribute encoding the formatting type, e.g. *bold*, *italic*, *underline*, etc.

For the needs of automated text analysis the correct text flow must be encoded, i.e. which fragments form a continues text. Two problems appeared: paragraphs divided between pages and word hyphenation. Concerning the former, we assume that page break breaks block elements. Thus, a continuous paragraph according to the authors intention is divided into two **`<p>`** tags, a kind of 'technical' paragraphs. Both are joined with the help of TEI aggregation mechanism and the **`prev`** and **`next`** attributes that points to the previous and the next element.

Word hyphenation is an individual writing feature covered by punctuation analysis in the forensic linguistic. People use different marks for splitting words between lines (-,-/-, =, =/=). Reconstruction of split words is crucial for automatic text processing. TEI **`<hyph>`** tag is used to encode word hyphenation (as in the lexicons) and the hyphen occurrence, e.g. "*competi*`<hyph>-</hyph><lb/>`*tor*". A structure of a sample suicide note is presented below (we cannot present the original scan of the letter due to the law reasons).

```
<text>
  <body>
    <pb facs="0003.1-1.png" n="1"/>
    <opener>
      <p rend="step-left"><salute>MAMUSIU TATUSIU
      KOCHANY XXXXXXXX</salute></p>
    </opener>
    <p rend="step-center">PRZEPRASZAM WAS<lb/>
```

```
      ZA TO!</p>
    <p rend="center">KOCHAM WAS BARDZO</p>
    <closer>
      <p rend="center"><signed>WASZ XXXXXX</signed></p>
    </closer>
  </body>
</text>
```

## 4 Case Study

Division into sentences is important for forensic linguistics, however, due to the problematic recognition of sentences limits in the notes (e.g. defective punctuation), we decided to focus on lines and paragraphs only. The way of isolating paragraphs can be characteristic to a given writer [12], e.g., indentation or by other elements: a line, a symbol sequence, initials etc. We applied TEI **<ornament>** – horizontal line with type *line* for true horizontal lines (see Fig. 1, left) and type *characters* for string of elements (e.g. asterisks) (see Fig. 1, right).

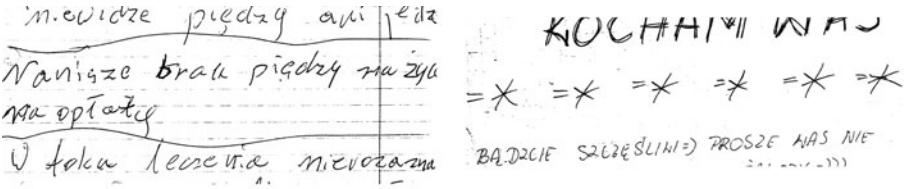

KOCHAM WAS
=* =* =* =* =* =*
BADZCIE SZCZĘŚLIWI =) PROSZE WAS NIE

**Fig. 1.** Examples of horizontal lines indicating new paragraphs: (a) (left) **<ornament type="line"/>** and (b) **<ornament type="characters"/>**

Writer's approach to hyphenation is an idiolectal feature. Authors very often avoid hyphenation that is associated with the central location of the text on the page and wide margins. Others hyphenate words in various ways, including mistakes related to syllabification ('podziekow/ać' *acknowledge*, 'pow/iedzenia' *saying*, 'króles-/stwa' *kingdom*). Wrong choice of punctuation mark or its position can be noticed, too:

- without punctuation mark ('nie/wygodni', encoded as *nie***<hyph/> <br/>***wygodni*);
- with hyphenation mark at the next line beginning ('hospita/-lizacja' *hospitalization*, as *hospitaliza***<br/><hyph>**-**</hyph>***cja*);
- with double hyphenation mark ('wspom-/-ę', as *wspom***<hyph>**-**</hyph><br/><hyph>**-**</hyph>***ę*);
- with equal sign ('chcia=/łem' wanted, encoded as *chcia***<hyph>**=**</hyph><br/>***łem*).

We extended TEI description of the hyphenation information about the usage of the punctuation mark and its location.

## 5 Corpus Statistics and Availability

*Polish Corpus of Suicide Notes* (PCSN) consists of 619 documents from years 1999–2008 obtained from prosecutor offices all over Poland. Demographical data of writers: male — 456, female — 160. The youngest authors are below 19 years old — 83 letters, the oldest were above 80 — 10 letters. Most of the suicide notes are handwritten — 604 letters, some are typed (computer, mobile phone) — 14 letters. Each note was scanned and transcribed. Fig. 2 presents the detailed statistics of corpus elements (state on the day of the 4th July 2011).

| Number of | Tag | Count |
|---|---|---|
| documents | `<body>` | 433 |
| pages | `<pb>` | 621 |
| paragraphs | `<p>` | 1767 |
| line breaks | `<lb/>` | 5427 |
| envelopes | `<envelope>` | 17 |
| letter openers | `<opener>` | 164 |
| letter closers | `<closer>` | 240 |

| Number of | Tag | Count |
|---|---|---|
| post scriptums | `<ps>` | 84 |
| signatures | `<signed>` | 197 |
| corrections | `<corr>` | 2996 |
| figures | `<figure>` | 104 |
| hyphenations | `<hyph>` | 90 |
| ornaments | `<ornament>` | 64 |

**Fig. 2.** Statistics of major elements in CPNS

The corpus will be available to other researchers on the basis of a free research license after signing an appropriate agreement. More information about the license conditions will be published on the PCSN web page: `http://pcsn.uni.wroc.pl`.

## 6 Conclusions and Further Research

Suicide notes are short texts and a significant part of information is expressed by the note visual structure or graphical symbols. Thus, a rich annotation scheme was proposed on the basis of TEI standard. The described aspects were divided generally into structural and related to the note content. The description of the note logical structure follows TEI adaptation for handwritten letters. As author-generated language errors can be important feature of information concerning the given individual, errors and their correction received especial attention. The proposed annotation scheme was tested on the basis of selected transcribed documents of different types. This work is continued. We plan to extend the corpus annotation with linguistic features in a semi-automated way: first applying language tools (e.g. a morpho-syntactic tagger or Named Entity recogniser) and next correcting the results manually.

**Acknowledgements.** Research paper financed partially with funds earmarked for research project in the budget for 2010–2012.

## References

1. Blackwell, S.: Why Forensic linguistics Needs Corpus Linguistics. Comparative Legilinguistics 1, 5–19 (2009)
2. Shneidman, E.S., Farberow, N.L. (eds.): Clues to Suicide, New York-Toronto-London (1957)
3. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie (eds.): The General Inquirer: A Computer Approach to Content Analysis, pp. 527–535. MIT Press, Cambridge (1969)
4. Jones, N.J., Bennell, C.: The Development and Validation of Statistical Prediction Rules for Discriminating Between Genuine and Simulated Suicide Notes. In: IASR, vol. 11, p. 230 (2007)
5. Pestian, J., Nasrallah, H., Matykiewicz, P., Bennett, A., Leenaars, A.: Suicide Note Classification Using Natural Language Processing: A Content Analysis, pp. 19-28, http://www.la-press.com
6. Eisenwort, B., Berzlanovich, A., Willlinger, U., Eisenwort, G., Lindorfer, S., Sonneck, G.: Abschiedsbriefe und ihre Bedeutung innerhalb der Suizidologie. Nervenarzt 77, 1359 (2006)
7. Olsson, J.: Wordcrime. Solving Crime Through Forensic Linguistics, London - New York, p. 55 (2009)
8. Razak, Z., Zulkiflee, K., Idris, M.Y.I., Tamil, E.M., Noor, M.N.M., Salleh, R., Yaakoob, M., Yusof, Z.M., Yaacob, M.: Off-line Handwriting Text Line Segmentation: A Review. In: IJCNS, vol. 8(7), p. 12 (2008)
9. Bosma, W., Vossen, P., Soroa, A., Rigau, G., Tesconi, M., Marchetti, A., Monachini, M., Aliprandi, C.: KAF: a generic semantic annotation format. In: Proc. of the 5th Inter. Conf. on Generative Approaches to the Lexicon GL 2009, Pisa, Italy (2009)
10. Vanhoutte, E., Van den Branden, R.: Describing: Transcribing, Encoding, and Editing Modern Correspondence Material: A Textbase Approach. Lit Linguist Computing 24(1), 77–98 (2009)
11. Coulthard, M.: On the Use of Corpora in the Analysis of Forensic Texts. Int. J. Speech Lang. La. 1, 27–43 (1994)
12. Olsson, J.: Forensic Linguistics. In: An Introduction to Language, Crime and Law, London - New York, p. 52 (2007)

# Unsupervised Russian POS Tagging with Appropriate Context

Li Yang[1], Erik Peterson[1], John Chen[1], Yana Petrova[2], and Rohini Srihari[1]

[1] Janya Inc.
1408 Sweet Home Road, Suite 1
Amherst, NY 14228, USA
{lyang,epeterson,jchen,rohini}@janya.com
[2] Department of Linguistics
State University of New York at Buffalo
Buffalo, NY 14260, USA
petrova3@buffalo.edu

**Abstract.** While adopting the contextualized hidden Markov model (CHMM) framework for unsupervised Russian POS tagging, we investigate the possibility of utilizing the left, right, and unambiguous context in the CHMM framework. We propose a backoff smoothing method that incorporates all three types of context into the transition probability estimation during the expectation-maximization process. The resulting model with this new method achieves overall and disambiguation accuracies comparable to a CHMM using the classic backoff smoothing method for HMM-based POS tagging from [17].

**Keywords:** unsupervised Russian part-of-speech tagging, CHMM, left, right, and unambiguous context, transition probability, expectation-maximization (EM).

## 1 Introduction

A careful review of the work on unsupervised POS tagging in the past two decades reveals that the hidden Markov model (HMM) has been the standard approach since the seminal work of [12] and [14] and that researchers sought to improve HMM-based unsupervised POS tagging from a variety of perspectives, including exploring dictionary usage, context utilization, sparsity control and modeling, and parameter and model updates tuned to linguistic features. For example, [3] and [7] utilized contextualized HMM (CHMM) to capture rich context. To account for sparsity, [8] and [10] utilized the Dirichlet hyperparameters of the Bayesian HMM. [4] integrated the discriminative logistic regression model into the M-step of the standard generative model to allow rich linguistically-motivated features.

Unsupervised systems went beyond the mainstream HMM framework by employing methods such as prototype-driven clustering [9,1], Bayesian LDA [18], integer programming [16], and K-means clustering [13].

Despite this large body of work, little effort has been devoted to unsupervised Russian POS tagging. Supervised Russian POS systems emerged in recent years. For example, eleven supervised systems entered the POS track of the 2010 Russian Morphological

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 427–433, 2011.

Parsers Evaluation[1]. Although the top two systems from the 2010 Evaluation achieved near perfect accuracy over the Russian National Corpus, little has been done on unsupervised Russian POS tagging. In this paper, we present our solution to unsupervised Russian POS tagging by adopting the CHMM. Our choice is based on the accuracy and efficiency of CHMM, an identical rationale to that behind [7].

We aim to achieve two goals. First, we intend to resolve the potential issue of missing useful contextual features by the backoff smoothing scheme in [17] and [7] for transition probabilities. Second, we explore the possibility of incorporating unambiguous context into transition probability estimation in an HMM framework. We propose a novel plan to achieve both goals in a unified approach.

In the following, we adopt the CHMM for unsupervised Russian POS tagging in section 2. Section 3 highlights the potential issue of missing useful left context in the backoff scheme by [17]. Section 4 illustrates an updated backoff scheme to resolve this potential issue. This scheme also unifies the left, right, and unambiguous context. The experiments and discussion are presented in section 5. We present conclusions in section 6.

## 2 CHMM for Russian POS Tagging

Our system is built upon the architecture of a contextualized HMM. Like other existing unsupervised HMM-based POS systems, the task of unsupervised POS tagging for us is to construct an HMM to predict the most likely POS tag sequence in the new data, given only a dictionary listing all possible parts-of-speech of a set of words and a large amount of unlabeled text for training.

Traditionally, the transition probability in a second-order HMM is given by $p(t_i|t_{i-2}t_{i-1})$, and the emission probability by $p(w_i|t_i)$ ([11,3]). The CHMM, such as [3], [2], and [7], incorporates more context into the transition and emission probabilities. Here, we adopt the transition probability $p(t_i|t_{i-1}t_{i+1})$ of [2] and [7] and the emission probability $p(w_i|t_it_{i+1})$ of [2].

Our training corpus consists of all 406,342 words of the plain text for training from the Appen Russian Named Entity Corpus[2], containing textual documents from a variety of sources. We created a POS dictionary for all 61,020 unique tokens in this corpus, using the output from the Russian lemmatizer [3]. The lemmatizer returns the stems of words and a list of POS tags for each word, relying on the morphology dictionary of the AOT Team [4]. Our tag set consists of 17 tags, comparable to those [5] used in Russian National Corpus (RNC), with the only addition of the *Punct* tag for punctuation marks. We relied on the Appen data because we did not have access to the RNC when our project was being developed. But we hope to be able to train and test out system with the RNC in the future.

[1] See http://ru-eval.ru/tables_index.html

[2] Licensed from http://www.appen.com.au/

[3] Available at http://lemmatizer.org/en/

[4] See http://aot.ru/

[5] Listed at http://www.ruscorpora.ru

## 3 Parameter Estimation and a Potential Issue

Given the model and resources for training described in section 2, we estimate the model parameters for our CHMM by following the standard EM procedures. During pre-processing, the dictionary is consulted, and a list of potential POS tags is provided for each word/token in the training sequence. In case of unknown words, the morphology analyzer built in the Russian lemmatizer suggests a list of tags. If the morphology analyzer does not make any suggestion, a list of open POS tags are assigned to the unknown words.

The potential POS tags in the training data provide counts to roughly esitimate the initial transition and emission probabilities. [2] initialized transition probabilities using a small portion of the training data. In our work, we initialize the emission probabilities using $20\%$ of the training data with $p(w_i|t_i t_{i+1}) = \frac{\#(w_i, t_i, t_{i+1})}{\#(t_i, t_{i+1})}$. During the EM process, we use additive smoothing when estimating $p(w_i|t_i t_{i+1})$ [6].

We initialize the transition probabilities $p(t_i|t_{i-1}t_{i+1})$ with a uniform distribution. When re-estimating $p(t_i|t_{i-1}t_{i+1})$, we use the method from [17] for backoff smoothing in equation (1).

$$\hat{p}(t_i|t_{i-1}t_{i+1}) = \lambda_3 \frac{N_3}{C_2} + (1-\lambda_3)\lambda_2 \cdot \frac{N_2}{C_1} + (1-\lambda_3)(1-\lambda_2) \cdot \frac{N_1}{C_0} \tag{1}$$

The $\lambda$ coefficients are calculated the same way as in [17], that is $\lambda_2 = \frac{\log(N_2+1)+1}{\log(N_2+2)}$ and $\lambda_3 = \frac{\log(N_3+1)+1}{\log(N_3+2)}$. The counts, $N_i$ and $C_j$ are modified for our unsupervised CHMM, as shown in Table 1. Note that $N_2$ captures the counts of the bi-gram $t_i t_{i+1}$, consisting of the current state $t_i$ and its right context $t_{i+1}$.

[17] and [7] show that equation (1) is quite effective in both supervised and unsupervised scenarios. However, in our case where Russian is concerned, there are situations where equation (1) may not give good estimates.

Through RNC's online search tool, we discovered that the word from a specific set of pronouns following the comma is always analyzed as a conjunction, which itself can be followed by a number of possible POS tags. This set includes ambiguous words such as *chto* and *chem*. Although the Appen corpus does not come with POS tags, our Russian linguist observed similar linguistic regularties in the corpus. Some examples regarding *chto* from Appen are listed below.

**Example 1** *,(Punct) chto(CONJ) na(PREP)*
**Gloss** comma and/or/that on

**Table 1.** Estimated counts, marked by superscript $^e$

| | |
|---|---|
| $N_1 = N_1^e$ | estimated counts of $t_{i+1}$ |
| $N_2 = N_2^e$ | estimated counts of $t_i t_{i+1}$ |
| $N_3 = N_3^e$ | estimated counts of $t_{i-1} t_i t_{i+1}$ |
| $C_0 = C_0^e$ | estimated total $\#$ of tags |
| $C_1 = C_1^e$ | estimated counts of $t_i$ |
| $C_2 = C_2^e$ | estimated counts of $t_{i-1} t_{i+1}$ |

**Example 2** *,(Punct) chto(CONJ) gotovy(ADJ)*
**Gloss** comma and/or/that ready

In the preceding examples, the comma to the left of *chto* provides for a useful clue. However, a potential issue arises when we estimate $p(t_{i-1}t_it_{i+1})$ using equation (1). That is, when the trigram $t_{i-1}t_it_{i+1}$ is rare and the first term of the equation is very small, the second term will affect $\hat{p}(t_{i-1}t_it_{i+1})$ more. The count, $N_2$, in the second term is for the bi-gram (*chto-CONJ, right word-POS*) but not for (*left word-comma, chto-CONJ*). Therefore, the useful clue in the latter bi-gram is missed. To resolve this, one cannot simply switch to the left context in $N_2$ because there are cases where the right context provides more of a clue. For example, observed from the Russian National Corpus, adjectival pronouns are only followed by a noun or an adjective and a noun, where the right context of adjectival pronouns are more important for disambiguating the adjectival pronouns. Several more examples from the Appen data where the left or right context contributing to disambiguation are listed in the Appendix.

## 4 Incorporating All Three Types of Context

Several systems made use of the information provided in unambiguous POS tag sequence. [5] learned rules from the context of unambiguous words. [15] created equivalence classes from unambiguous words for training. We expected the assumption that unambiguous context helps with disambiguation to hold for Russian as well.

In the Appen training corpus, $84\%$ of the words/tokens have a unique POS tag, based on our dictionary and the Russian lemmatizer. We can easily spot examples in the corpus where unambiguous context helps with disambiguation. Again, in our earlier example, *,(Punct) chto(CONJ) na(PREP)*, the unambiguous left context ',' reveals that *chto* is a CONJ instead of a PRON. To take advantage of the unambiguous context, we collect the counts for all unambiguous tri-gram and bi-gram sequences in the Appen training corpus and integrate these counts into equation (2) through the equivalence in Table 2.

$$\hat{p}(t_i|t_{i-1}t_{i+1}) = \lambda_3\frac{N_3}{C_2} + (1-\lambda_3)\lambda_2 \cdot \frac{N_2^L}{C_1^L} \times \frac{N_2^R}{C_1^R} + (1-\lambda_3)(1-\lambda_2)\cdot\frac{N_1}{C_0} \quad (2)$$

where $\lambda_2 = \frac{\log(N_2^L+1)+1}{\log(N_2^L+2)} \times \frac{\log(N_2^R+1)+1}{\log(N_2^R+2)}$, and $\lambda_3 = \frac{\log(N_3+1)+1}{\log(N_3+2)}$. $\lambda_2$ incorporates both the left and right context. The unambiguous counts are defined in Table 2.

**Table 2.** Counts of unambiguous tri-grams, bi-grams, and unigrams. The superscript $^u$ stands for unambiguous counts.

| |
|---|
| $N_1 = N_1^u$ , # of unambiguous counts of $t_{i+1}$ |
| $N_2^L = N_2^{uL}$ , # of unamb. bi-gram $t_{i-1}t_i$ $w$ left context $t_{i-1}$ |
| $N_2^R = N_2^{uR}$ , # of unamb. bi-gram $t_it_{i+1}$ $w$ right context $t_{i+1}$ |
| $N_3 = N_3^u$ , # of unamb. tri-gram $t_{i-1}t_it_{i+1}$ |
| $C_0 = C_0^u$ , total # of unamb. tags |
| $C_1 = C_1^u$ , # of unamb. $t_i$ |
| $C_2 = C_2^u$ , # of unamb. bi-gram of $t_{i-1}t_{i+1}$ |

Now that the new backoff smoothing plan combines both the left and right unambiguous bi-gram counts, we extend this plan to cover the cases where the unambiguous tri/bi/uni-grams are not available, by replacing them with the estimated counts from Table 1. Table 3 displays the scheme for replacing an unambiguous count with an estimated count from the EM process.

**Table 3.** Replacement plan for unambiguous counts

| | |
|---|---|
| $N_1^u \leftarrow N_1^e$ | estimated counts of $t_{i+1}$ |
| $N_2^{uL} \leftarrow N_2^{eL}$ | estimated counts of $t_{i-1}t_i$ |
| $N_2^{uR} \leftarrow N_2^{eR}$ | estimated counts of $t_it_{i+1}$ |
| $N_3^u \leftarrow N_3^e$ | estimated counts of $t_{i-1}t_it_{i+1}$ |
| $C_0^u \leftarrow C_0^e$ | estimated total $\#$ of tags |
| $C_1^u \leftarrow C_1^e$ | estimated counts of $t_i$ |
| $C_2^u \leftarrow C_2^e$ | estimated counts of $t_{i-1}t_{i+1}$ |

## 5 Experiments and Results

We designed three experiments to test three combinations of the context, in addition to experimenting with a traditional second-order HMM. The Appen corpus contains a development set and an evaluation set. We passed both sets through the Russian lemmatizer to obtain POS tags for the data and had the tags manually corrected by a Russian linguist. Thus, we have created both development and evaluation data. $14\%$ of words/tokens in both development and evaluation data have multiple POS tags. Table 4 summarizes our experimental settings and results over the evaluation data.

**Table 4.** Experiments, overall and disambiguation accuracies over test data

| Model & setting(s) | Overall Accuracy | Disamb. Accuracy |
|---|---|---|
| 2nd-order HMM | $94.88\%$ | $63.42\%$ |
| CHMM_left_context | $95.72\%$ | $69.42\%$ |
| CHMM_right_context | $96.05\%$ | $71.78\%$ |
| CHMM_unique_<br>$\leftarrow$_left/right context | $96.06\%$ | $71.85\%$ |

The second-order HMM was trained with the traditional transition probability $p(t_i|t_{i-2}t_{i-1})$ and emission probability $p(w_i|t_i)$. It gained an overall accuracy of $94.88\%$, and was able to correctly disambiguate $63.42\%$ of the ambiguous words/tokens.

All three CHMM models were trained with the emission probability $p(w_i|t_it_{i+1})$ initialized with $20\%$ of the unlabeled training corpus. Model *CHMM_left_context* considered the left context bi-gram $t_{i-1}t_i$ when calculating the second term in equation (1). Model *CHMM_right_context* considered the right context bi-gram $t_it_{i+1}$ when calculating the same term. Model *CHMM_unique_* $\leftarrow$ *_left/right* unified both unambiguous context counts and estimated counts for left and right context from the EM process, using equation (2).

All CHMM models achieved accuracies $1\%$ higher than the HMM, while the disambiguation accuracies from the former three are $7-9\%$ higher than the latter. This shows that the CHMM models capture more useful context information for Russian POS tagging than the traditional HMM. At the same time, the overall and disambiguation accuracies between *CHMM_right_context* and *CHMM_unique_* ← *_left/right* are comparable. Error analyses indicate that a backoff scheme for emission probabilities is also needed to incorporate the left context.

## 6 Conclusion and Future Work

We adopted the CHMM to unsupervised Russian POS tagging. The CHMM models using either the left or right context were able to outperform the traditional second-order HMM. To resolve the potential issue of missing out the left context with the classic smoothing scheme in [17], we experimented with an approach to unifying the information provided in the left, right, and unambiguous contexts. The results from the latter were comparable to a CHMM with the classic backoff smoothing method in [17], although we expected a more significant improvement. We plan to investigate a backoff scheme for emission probabilities where we will incorporate the left context as well, while currently we only rely on additive smoothing for emission probabilities.

**Acknowledgments.** We would like to thank the anonymous reviewers for their valuable comments and suggestions. Our work was partially funded by the Air Force Research Laboratory/RIEH in Rome, New York through contracts FA8750-09-C-0038 and FA8750-10-C-0124.

## References

1. Abend, O., Reichart, R., Rappoport, A.: Improved unsupervised pos induction through prototype discovery. In: Proceedings of the 48th ACL (2010)
2. Adler, M.: Hebrew Morphological Disambiguation. Ph.D. thesis, University of the Negev (2007)
3. Banko, M., Moore, R.C.: Part of speech tagging in context. In: Proceedings of the 20th International Conference on Computational Linguistics (2004)
4. Berg-Kirkpatrick, T., Bouchard-Ct, A., DeNero, J., Klein, D.: Painless unsupervised learning with features. In: Proceedings of NAACL 2010 (2010)
5. Brill, E.: Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging. In: Very Large, pp. 1–13. Kluwer Academic Press, Dordrecht (1995)
6. Chen, S.F.: Building Probabilistic Models for Natural Language. Ph.D. thesis, Harvard University (1996)
7. Goldberg, Y., Adler, M., Elhadad, M.: Em can find pretty good pos taggers (when given a good start). In: Proceedings of ACL 2008: HLT (2008)
8. Goldwater, S., Griffiths, T.: A fully bayesian approach to unsupervised part-of-speech tagging. In: Proceedings of the 45th ACL (2007)
9. Haghighi, A., Klein, D.: Prototype-driven learning for sequence models. In: Proceedings of the main conference on HLT-NAACL (2006)

10. Johnson, M.: Why doesnt em find good hmm pos-taggers. In: n EMNLP (2007)
11. Kriouile, A.: Some improvements in speech recognition algorithms based on hmm. In: Acoustics, Speech, and Signal Processing (1990)
12. Kupiec, J.: Robust part-of-speech tagging using a hidden markov model. Computer Speech & Language 6, 225–242 (1992)
13. Lamar, M., Maron, Y., Bienenstock, E.: Latent descriptor clustering for unsupervised pos induction. In: EMNLP 2010 (2010)
14. Merialdo, B.: Tagging english text with a probabilistic model. Computational Linguistics 20, 155–171 (1994)
15. Mihalcea, R.: The role of non-ambiguous words in natural language disambiguation. In: Proceedings of the Conference on RANLP (2003)
16. Ravi, S., Knight, K.: Minimized models for unsupervised part-of-speech tagging. In: Proceedings of ACL-IJCNLP 2009, pp. 504–512 (2009)
17. Thede, S.M., Harper, M.P.: A second-order hidden markov model for part-of-speech tagging. In: Proceedings of the 37th Annual Meeting of the ACL (1999)
18. Toutanova, K., Johnson, M.: A bayesian lda-based model for semi-supervised part-of-speech tagging. In: Proceedings of NIPS (2007)

## Appendix: Linguistic Patterns Observed in Appen

In Section 3, we illustrated how the left context helped to disambiguate *chto*. In the following we present several more examples from the Appen corpus illustrating the helpful left or right context. While the patterns our Russian linguist observed are common in both the RNC and Appen, the counts and statistics regarding each pattern are unavailable for reporting because the RNC was then inaccessible to us and Appen was not tagged with POS tags.

Examples 3 through 7 show that the left context of *chem*, *poka*, and *kak* helps to disambiguate them as conjuctions.

**Example 3** *,(Punct) chem(CONJ) v(PREP) stolitse(NOUN)*
**Gloss** comma and/than in capital
**Example 4** *,(Punct) poka(CONJ) eta(PRONOUN)*
**Gloss** comma yet this
**Example 5** *,(Punct) poka(CONJ) Sovet(NOUN)*
**Gloss** comma yet council
**Example 6** *,(Punct) kak(CONJ) dva(NUMERAL) neudachnika(NOUN)*
**Gloss** comma as two losers
**Example 7** *,(Punct) kak(CONJ) on(PRONOUN)*
**Gloss** comma as he

The next examples show that the right context determines the adjectival tag, *PRONOUN_P*, of the pronouns.

**Example 8** *obekty(NOUN) svoey(PRONOUN_P) sistemy(NOUN)*
**Gloss** units their/they system
**Example 9** *esli(CONJ) mnogie(PRONOUN_P) mnogie(NOUN)*
**Gloss** if many/various emigrants

# WCCL: A Morpho-syntactic Feature Toolkit*

Adam Radziszewski, Adam Wardyński, and Tomasz Śniatowski

Institute of Informatics
Wrocław University of Technology
Wybrzeże Wyspiańskiego 27,
Wrocław, Poland

**Abstract.** The paper presents WCCL, a new formalism and toolkit for constructing morpho-syntactic features, a crucial task for many natural language processing algorithms. One existing solution, JOSKIPI, is analysed from two perspectives: features of the formalism as well as software engineering-related issues. Then we propose its successor. A short case study follows, exemplifying the improvement enabled by using rich features expressed with WCCL. The formalism is targeted at Polish, although it seems well suited for any inflectional language.

## 1 Background

Many NLP tasks may be performed using Machine Learning (ML) methods. The bulk of knowledge extraction can be offloaded to an underlying model (e.g. a classifier, clusterer). Even so, some amount of knowledge engineering is still required, e.g., it is necessary to provide a set of *features*, such as properties of consecutive word forms or simple syntactic dependencies between word forms.

The construction of features is especially challenging for Slavic languages, where syntactic dependencies are often marked by morphological features rather than a particular ordering of word forms [9]. The importance of morpho-syntactic features is confirmed by experiments. For instance, a successful adaptation of Brill's tagging algorithm to inflectional languages includes a decomposition of tags into parts corresponding to sets of selected morpho-syntactic categories [1]. Similarly, explicit tests for adjective–noun morphological agreement improved semantic similarity measures extracted from large corpora [7].

The ability to provide good morpho-syntactic features is an important factor in the construction of NLP systems. Furthermore, different tasks, such as morpho-syntactic tagging and lexico-semantic relation extraction, refer to similar types of information: parts-of-speech, values of morpho-syntactic categories, simple syntactic dependencies such as grammatical agreement. These observations suggest that a common reusable framework for expressing such features could contribute to a productivity increase in the NLP community.

There exists one formalism (with two implementations) that fits into the described application scenarios, namely the JOSKIPI language [6,8]. In the next section we argue that although JOSKIPI is conceptually well-founded, the design and implementation

* This work is financed by Innovative Economy Programme project POIG.01.01.02-14-013/09.

I. Habernal and V. Matoušek (Eds.): TSD 2011, LNAI 6836, pp. 434–441, 2011.

flaws render the software not applicable as a general feature extraction toolkit. Then we propose our solution: an extended formalism for expressing morpho-syntactic features, constraints and tagging rules, as well as its implementation — a toolkit designed to be used as a reusable component in language engineering.

## 2 JOSKIPI

JOSKIPI originates from TaKIPI, a morpho-syntactic tagger for Polish [6]. Detailed description of the formalism can be found in [8].

The formalism was devised for writing rules for the tagger, as well as a means to define basic functional expressions that would be building blocks for automatically extracted rules. The functional expressions (called *operators*) operate on sentences, understood as sequences of morphologically analysed (but possibly not fully disambiguated) tokens, with one token marked as the *current position*. Operators refer to tokens using position relative to the analysed centre: 0 is the current position, -1 is the first token positioned to the left, 1 is the first token to the right etc.; e.g. the operator `cas[-1]` returns the value of grammatical case for the token preceding the current position. The current position is moved sequentially through the sentence and the operators are evaluated every time.

Although such operators may be used to identify simple syntactic dependencies, JOSKIPI is not a shallow parser. The assumptions are different: a shallow parser captures and labels sequences of tokens, while JOSKIPI enables to evaluate functional expressions against given context. The values returned are to support decision making, whether the task is segmentation, disambiguation or relation finding.

### 2.1 Implementations and Applications

There exist two implementations of JOSKIPI (both available under GNU GPL):

1. The original implementation, a part of the TaKIPI tagger [6]. The implementation is provided as a C++ shared library with limited support for command-line processing.
2. An experimental Python implementation, a part of the Disaster system [12]. The original language is extended with means of referring to shallow syntactic annotation and features such as explicit variable assignment, regular expressions and boolean literals.

The language proved useful for several different applications. JOSKIPI predicates were used to filter a list of candidate Multi-Word Expressions acquired automatically [8]. Furthermore, JOSKIPI operators were used to generate features for extraction of lexico-semantic relations as performed in the SuperMatrix system [2]. The Python version was used for shallow parsing of Polish text [12].

### 2.2 Limitations and Design Flaws

Despite the successful applications, we argue that neither of the implementations is applicable as a general toolkit for feature generation due to design and implementation flaws. The situation is worsened by the lack of technical documentation and the poor design of the available APIs.

The C++ implementation uses a hard-coded tagset definition, citing computational efficiency reasons. Unfortunately this causes even slight modifications to be labour-intensive, since one tagset attribute or value is defined in several places in the source code. This also makes the project not likely to survive for a longer time in the face of the changing standards, e.g. when the announced National Corpus of Polish with its own tagset [10] becomes available.

The language specification lacks strict data type definition. Even if an operator only makes sense when supplied with strings, it can be applied to a symbol set or even a boolean value. It is not possible to infer the type of an operator, or the type of a returned value. Strings are kept as indices to a global string table, which hinders distributed computation and is prone to memory leaks.

The Python implementation on the other hand is far too slow to be useful for processing large corpora. For instance, applying the expression `cas[0]` (i.e. retrieving the value of the grammatical case) to a 800 000-token corpus takes about 3.5 minutes on a 2.2 GHz Athlon 64 PC (around 4000 tokens per second).

## 3 Proposed Successor to JOSKIPI

We wanted to provide a toolkit that would capitalize on strengths of JOSKIPI formalism and fix its shortcomings. The new, redesigned and extended version of the feature description language is what we have called WCCL (for Wrocław Corpus Constraint Language). The implementation is written in C++ as a shared library, using and extending the Corpus2 library from the Maca system [13] for basic data structures and I/O.

### 3.1 Toolkit Features

The language supports configurable tagset read from definitions similar to those in [1]. Thus, it is not bound to any particular language. Configuring WCCL with various tagset definition works seamlessly with parts of the language that use tagset symbols. For example, if the employed tagset contains an attribute `tense` with three values: `pres`, `past`, `fut`, then `equal(tense[0], {pres})` is a valid predicate that checks whether the current token's `tense` is `pres`.

Variables in a parsed expression are available at run-time for inspection and assignment. This is especially useful in the case of string variables: it is possible to create only one operator instance and then manipulate the variables; in JOSKIPI it was necessary to wastefully create thousands of operators.

As JOSKIPI lacked a convenient means of referring to lexical information, we extended the formalism with possibility to refer to external lexicon files. Lexicons are essentialy lists of key–value pairs. They may be used by special `lex` operator to translate sets of strings into the corresponding values. Items not present in the lexicon are omitted in the output, so this mechanism can be used both for classification and for filtering of infrequent forms.

While not suitable for feature selection, disambiguation rules that were available in JOSKIPI are also present in WCCL — making it effectively a rule-based disambiguation engine with configurable tagset.

We have created a clear specification of WCCL file syntax. Whole file parsing enables users to put operators in named groups and subsequently access them in a convenient fashion. This directly supports creation of feature sets for classification. The syntax also allows a header for lexicon imports and a section for tagging rules.

Support for parallel processing is provided. The operators have no state apart from a well-separated container for variables, and as such a complex operator can be shared between threads, potentially reducing memory use.

WCCL also provides usable command-line utilities, e.g., for evaluation of multiple operators against a corpus (producing feature values). The utilities may be used in tandem with Maca analysis and conversion utilities through input-output piping. This way we achieve the architectural model of maximal component decoupling that is recommended for NLP toolkits [5]. In addition, the API is available both as a native C++ code and simple Python wrappers that enable rapid NLP application development and fast prototyping.

### 3.2 WCCL Operators

WCCL is strongly-typed: all expressions have a well-defined type. The range of available types currently consists of positions (integers), string sets, boolean values and tagset symbol sets. Variables of all the types are supported, in contrast to JOSKIPI, which supported only variables of the position type. Each type has a defined syntax for variable definition and literals, which enables automatic type inference in functions such as `equal`. The subsequent enhancements include an explicit functional-style *if* statement with inferable type.

WCCL operators work in similar fashion to the aforementioned JOSKIPI operators. To an extent, they are a superset of JOSKIPI functionality as we did want existing rules to be easily applicable to the new, extended formalism. Below we enumerate types of operators in WCCL.

1. Constants for all types.
   (a) Configurale tagset symbols: symbols of grammatical classes (roughly, parts-of-speech) and values of grammatical categories as defined in given tagset (e.g. the tagset of the IPI PAN Corpus (IPIC) [11]). Examples of such operators include: `{}` (empty set); `{nmb}` (*number* category, equivalent to `{sg, pl}` - *singular* and *plural*); `{f, n}` (values of *feminine* and *neuter* from the *gender* category, excluding masculine values)
   (b) Set of strings, e.g. `[]` (empty set); `"water"` (single string; equivalent to `["water"]`); `["ice", "water"]` (constant string set with two values)
   (c) Position, e.g. `begin` (begining of a sentence); `end` (end of a sentence); `0` (current position); `-2` (position second to the left from current position)
   (d) Boolean, `True` or `False`
2. Retrieving value of variables for all types, with names prefixed according to the type, e.g. `$Pos` (position variable named *Pos* - no prefix for this type); `$s:S`

(string set variable named *S*); `$t:T` (variable *T* for a set of tagset symbols); `$b:F` (boolean variable *F*).

3. Retrieving values of morpho-syntactic categories. Such operators are defined for the grammatical class and all the tagset attributes with names depending on the tagset given. They take a position as an argument and return set of values for given category from all lexemes of a token pointed by the position. E.g.: `cas[0]` (returns value of *case* category); `class[$V]` (returns word class of token pointed by position variable named *V*).
4. Retrieving string values: lemmas or orthographic forms. Again, as lemmas may be ambiguous, sets are returned. E.g: `orth[0]` (orthographic form of token at current position); `base[2]` (lemmas for the second token to the right from the current one).
5. Simple predicate constraints that allow testing whether values returned by various operators satisfy a relation. E.g. `equal(orth[0], "ice")` (is the orthographic form of the current token equal to *ice*?); `inter(gnd[-1], {n, f})` (does the set of values for *gender* category taken from the token preceding the current one intersect with constant set `{n, f}`?).
6. Constraints that test for morpho-syntactic agreement on a given set of grammatical categories between two specified tokens or a range of tokens.
7. Search operators that try to find a token satisfying a constraint in an iteration-like fashion using variables. The operators mimic regular predicates and can be used in conjunction with functional expressions on the token found.
8. Logical predicate operators (`and`, `not`, `or`) and conditional operator (`if`) that allow composition of simpler operators to create more complex ones.
9. Aforementioned `lex` operator that translates a set of strings according to a lexicon read from an external file.

An example operator utilising the language extensions is presented below.

```
if(
   rlook(0, end, $Pos, inter($s:Lemma,base[$Pos])),
   class[$Pos], // return the grammmatical class if found
   {ign} // else clause
)
```

The operator examines the sentence, starting from the central position (`0`), proceeding left-to-right (the iteration is realised by increasing the value of the `$Pos` variable). The first token whose possible lemma set intersects (`inter` predicate) with the string set given via an external variable (`$s:Lemma`) is taken, and its grammatical class returned. If no token satisfying the condition is found, the value of the *else* clause is taken — in this case, `ign` grammatical class (unknown form in the IPIC tagset).

## 4 Case Study: Creating a Memory-Based Chunker

Chunking is the task of identifying phrase boundaries in text. *NP chunking* is limited to recognising noun phrases (NPs). When using ML techniques, the usual practice is to treat the task as a sequence labelling problem, i.e., special tags that denote whether a token is inside, outside or begins an NP are attached to tokens [14]. The task was performed for Polish using decision trees and hand-tailored features, yielding 85.7% precision and 84.9% recall [12].

Memory-Based Learning (MBL) is an ML technique where instead of the usual generalisation during training, all the training examples are memorised. The actual classification is based on finding a number of most similar examples in the memory and retrieving their class label. MBL has been successfully applied to various NLP tasks including chunking [3]. We can attempt to create a chunker for Polish using MBT, a software toolkit for memory-based tagging [4], to examine the impact of additional features generated by WCCL in a real NLP task.

The baseline was to train and test MBT on input consisting of the word forms, morpho-syntactic description (MSD) tags and the corresponding chunk tags (class labels). MBT treats the feature values atomically, hence tags are not decomposed into parts in this setting.

More sophisticated features were introduced by writing WCCL expressions. The first improvement was to provide features for grammatical class, number and gender in a fixed window $(-3,\ldots,2)$ (similar features are used in TaKIPI [6]). As the NP chunks that we try to recognise are based on the agreement on number, gender and case, we introduced explicit tests for such agreements as the second improvement by providing the following WCCL predicates:

1. checking for agreement on two- and three-token ranges crossing the current position (5 predicates),
2. checking for "weak" agreement[1] on a token range that stretches between an adjective and a noun with those boundaries possibly being several tokens away from the current position (for both possible word orders).

The third step involved enriching the lexical information. In this setting we added 5 additional features accounting for the wordforms of all the tokens in the fixed window. To avoid inclusion of infrequent forms into the domain, we created a frequency list from an extra part of the employed corpus[2] and created a lexicon of 800 most frequent wordforms. The lexicon was then used by WCCL expressions that kept the encountered forms intact if present in the lexicon, while mapping the rest to empty symbol (e.g. `lex(lower(orth[0]), "freq")`).

The features for the third set-up were generated using the following code (and the `wccl-run` utility):

```
import("800.txt", "freq") // import lexicon file as "freq"
@"simple" (
   class[-3]; nmb[-3]; cas[-3]; gnd[-3]; // repeated for -2..2
)
```

[1] Weak agreement is not violated when the range contains additional indeclinable forms.

[2] The chunk corpus [12] consists of a small chunk-annotated part (used for chunker evaluation) and a much larger part with morpho-syntactic annotation only. The latter was used to gather the frequency list.

```
@"lex" (
   lex(lower(orth[-3]), "freq"); // repeated for -2..2
)
@"agrs" (
   agrpp(-1,0,{nmb,gnd,cas}); agrpp(0,1,{nmb,gnd,cas});
   wagr(-2,0,{nmb,gnd,cas}); wagr(-1,1,{nmb,gnd,cas});
   wagr(0,2,{nmb,gnd,cas});
   if(and(
      llook(0,-2,$L,agrpp($L, $L, {nmb,gnd,cas})),
      rlook(0,4,$R,inter(class[$R], {subst,ger,depr})),
      wagr($L,$R, {nmb,gnd,cas})
   ), $R);
   if(and(
      rlook(0,2,$R,agrpp($R, $R, {nmb,gnd,cas})),
      llook(0,-4,$L,inter(class[$L], {subst,ger,depr})),
      wagr($L,$R, {nmb,gnd,cas})
   ),$L)
)
```

The results on the data set from [12] are reported in Tab. 1. The most impressive improvement is observerd after introduction of simple symbols extracted from tags. Adding of agreement tests as well as lexical information also helps. Our best results are still significantly worse than those achieved by [12], nevertheless it is instructive to see that an acceptable performance may be quickly achieved by using readily available components, and that the extra features improved the result significantly.

**Table 1.** The perfomance of MBL chunkers with different feature sets

| Chunker | Precision | Recall | F |
|---|---|---|---|
| MBL baseline: form, whole MSD tag | 0.4782 | 0.6267 | 0.5424 |
| MBL baseline + class, nmb, gnd, cas | 0.6708 | 0.7655 | 0.7150 |
| MBL baseline + class, nmb, gnd, cas, agreement | 0.7133 | 0.8008 | 0.7544 |
| MBL baseline + class, nmb, gnd, cas, agreement, lex | 0.7238 | 0.8039 | 0.7617 |
| Decision trees [12] | 0.8574 | 0.8490 | 0.8530 |

## 5 Conclusion

Providing features required by ML approaches in NLP can be simplified accross many NLP tasks if a proper toolkit is made available. We analysed JOSKIPI formalism along with its implementations and found it to be inadequate as a general purpose toolkit for processing Polish corpora. Taking it as a great starting point, we addressed the identified problems and added important missing features, such as support for different tagsets, broader type system with variables of every type, ability to assign values to variables at run time, support for parallel processing, and more.

We proposed a successor to JOSKIPI: the WCCL formalism and its implementation, a shared library and useful command-line utilites. We aim to additionally develop a web API — this way the formalism will be used as a corpus query language.

## References

1. Acedański, S., Gołuchowski, K.: A morphosyntactic rule-based brill tagger for polish. In: Proceedings of Intelligent Information Systems, pp. 67–76 (2009)
2. Broda, B., Piasecki, M.: SuperMatrix: a general tool for lexical semantic knowledge acquisition. In: Speech and Language Technology, vol. 11, pp. 239–254. Polish Phonetics Assocation (2008)
3. Daelemans, W., van den Bosch, A.: Memory-Based Language Processing. Cambridge University Press, Cambridge (2005)
4. Daelemans, W., Zavrel, J., van den Bosch, A., van der Sloot, K.: MBT: Memory-Based Tagger, version 3.2. Tech. Rep. 10-04, ILK (2010)
5. Leidner, J.: Current Issues in Software Engineering for Natural Language Processing. In: Patrick, J., Cunningham, H. (eds.) Proceedings of the HLT-NAACL 2003 Workshop (SEALTS), pp. 45–50 (2003)
6. Piasecki, M.: Polish tagger TaKIPI: Rule based construction and optimisation. Task Quarterly 11(1–2), 151–167 (2007)
7. Piasecki, M., Broda, B.: Semantic similarity measure of polish nouns based on linguistic features. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 381–390. Springer, Heidelberg (2007)
8. Piasecki, M., Radziszewski, A.: Morphosyntactic constraints in acquisition of linguistic knowledge for polish. In: Marciniak, M., Mykowiecka, A. (eds.) Bolc Festschrift, vol. 5070, pp. 163–190. Springer, Heidelberg (2009)
9. Przepiórkowski, A.: Slavic Information Extraction and Partial Parsing. In: Proceedings of the Workshop on Balto-Slavonic Natural Language Processing, pp. 1–10. ACL, Prague (2007)
10. Przepiórkowski, A.: A comparison of two morphosyntactic tagsets of Polish. In: Koseska-Toszewa, V., Dimitrova, L., Roszko, R. (eds.) Representing Semantics in Digital Lexicography: Proceedings of MONDILEX Fourth Open Workshop, Warszawa, pp. 138–144 (2009)
11. Przepiórkowski, A., Woliński, M.: A flexemic tagset for Polish. In: Proceedings of Morphological Processing of Slavic Languages, EACL 2003 (2003)
12. Radziszewski, A., Piasecki, M.: A preliminary noun phrase chunker for Polish. In: Proceedings of the Intelligent Information Systems (2010)
13. Radziszewski, A., Śniatowski, T.: Maca — a configurable tool to integrate Polish morphological data. In: Proceedings of FreeRBMT11 (2011)
14. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Proceedings of the Third ACL Workshop on Very Large Corpora, Cambridge, MA, USA, pp. 82–94 (1995)

# Author Index