Hrafn Loftsson
Eiríkur Rögnvaldsson
Sigrún Helgadóttir (Eds.)

Advances in Natural Language Processing

LNAI 6233

7th International Conference on NLP, IceTAL 2010
Reykjavik, Iceland, August 2010
Proceedings

Springer

Lecture Notes in Artificial Intelligence 6233
Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Hrafn Loftsson
Eiríkur Rögnvaldsson
Sigrún Helgadóttir (Eds.)

Advances in Natural Language Processing

7th International Conference on NLP, IceTAL 2010
Reykjavik, Iceland, August 16-18, 2010
Proceedings

Springer

Volume Editors

Hrafn Loftsson
School of Computer Science, Reykjavik University, IS-101 Reykjavik, Iceland
E-mail: hrafn@ru.is

Eiríkur Rögnvaldsson
Department of Icelandic, University of Iceland, IS-101 Reykjavik, Iceland
E-mail: eirikur@hi.is

Sigrún Helgadóttir
Arni Magnusson Institute for Icelandic Studies, IS-101 Reykjavik, Iceland
E-mail: sigruhel@hi.is

Library of Congress Control Number: 2010931612

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-14769-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-14769-2 Springer Berlin Heidelberg New York

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

Preface

The research papers in this volume comprise the proceedings of IceTAL 2010, an international conference on natural language processing (NLP). IceTAL was the seventh in the series of the TAL conferences, following GoTAL 2008 (Gothenburg, Sweden), FinTAL 2006 (Turku, Finland), EsTAL 2004 (Alicante, Spain), PorTAL 2002 (Faro, Portugal), VexTAL 1999 (Venice, Italy), and FracTAL 1997 (Besançon, France). The main goal of the TAL conference series has been to bring together scientists representing linguistics, computer science, and related fields, sharing a common interest in the advancement of computational linguistics and NLP. IceTAL 2010, organized by the Icelandic Centre for Language Technology (ICLT) in Reykjavík, Iceland, successfully contributed to these goals.

Our Program Committee (PC) consisted of 45 recognized researchers and professionals in the field of NLP from Belgium, China, Cuba, Denmark, Finland, France, Germany, Iceland, Italy, Japan, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom, and the United States.

We called for submissions both from academia and the industry on any topic that is of interest to the NLP community, particularly encouraging research emphasizing multidisciplinary aspects of NLP and the interplay between linguistics, computer science, and application domains such as biomedicine, communication systems, public services, and educational technology.

As a response, we received 91 submissions from authors representing 37 countries in Europe, Asia, Africa, Australia, and the Americas. Each submission was reviewed by three PC members or external reviewers designated by the PC members. The PC members as well as the external reviewers are gratefully acknowledged in the following pages for their valuable contribution. The reviewing process led to the selection of 43 papers (30 full papers and 13 short papers) to be presented at the IceTAL conference and published in this volume. We believe that the selected papers contribute significantly to the field of NLP and we hope that you will find them inspiring for your own NLP work.

We would like to express our gratitude to the institutions that collaborate through ICLT and made IceTAL possible: Reykjavik University, University of Iceland, and the Árni Magnússon Institute for Icelandic studies. Furthermore, we thank our invited speakers, the highly esteemed scholars Jan Hajič from Charles University in Prague, and Christiane D. Fellbaum from Princeton University in the United States. We also thank our sponsors, IZETeam, Microsoft Ísland, and the Post and Telecom Administration in Iceland, for their valuable support. Finally, we thank all the individuals that were involved in organizing IceTAL, without whom this event would not have been possible at all.

June 2010 Hrafn Loftsson

Organization

IceTAL 2010 was organized by the Icelandic Centre for Language Technology (ICLT) – an LT research, development and teaching platform between Reykjavik University, University of Iceland, and the Árni Magnússon Institute for Icelandic studies (AMI).

UNIVERSITY OF ICELAND

ÁRNI MAGNÚSSON INSTITUTE
FOR ICELANDIC STUDIES

Program Committee

Walid El Abed	Global Data Excellence Ltd., UK
Jan Alexandersson	DFKI, Germany
Jorge Baptista	University of Algarve, Portugal
Tilman Becker	DFKI, Germany
Chris Biemann	Powerset, USA
Kristín Bjarnadóttir	AMI, Iceland
Lars Borin	Gothenburg University, Sweden
Johan Bos	La Sapienza, Italy

Caroline Brun	Xerox Corporation, France
Sylviane Cardey	University of Franche-Comté, France
Robin Cooper	Gothenburg University, Sweden
Walter Daelemans	University of Antwerp, Belgium
Rodolfo Delmonte	University of Venice, Italy
Markus Dickinson	Indiana University, USA
Mikel L. Forcada	University of Alicante, Spain
Robert Gaizauskas	University of Sheffield, UK
Filip Ginter	University of Turku, Finland
Peter Greenfield	University of Franche-Comté, France
Philippe de Groote	INRIA Lorraine, France
Sigrún Helgadóttir	AMI, Iceland
Hitoshi Isahara	NICT, Japan
Janne Bondi Johannessen	University of Oslo, Norway
Krister Lindén	University of Helsinki, Finland
Hrafn Loftsson	Reykjavik University, Iceland (Chair)
Bente Maegaard	University of Copenhagen, Denmark
Sun Maosong	Tsinghua University, China
Leonel Ruiz Miyares	Centro de Lingüística Aplicada, Cuba
Joakim Nivre	Uppsala and Växjö University, Sweden
Pierre Nugues	University of Lund, Sweden
Guy Perrier	INRIA Lorraine, France
Liu Qun	Institute of Computing Technology, China
Aarne Ranta	Chalmers and Gothenburg University, Sweden
Eiríkur Rögnvaldsson	University of Iceland, Iceland
Tapio Salakoski	University of Turku, Finland
Karl-Michael Schneider	Cataphora, USA
Koenraad de Smedt	University of Bergen, Norway
Mark Stevenson	University of Sheffield, UK
Izabella Thomas	University of Franche-Comté, France
Trond Trosterud	University of Tromsø, Norway
José Luis Vicedo	University of Alicante, Spain
Simo Vihjanen	Lingsoft Ltd., Finland
Hannes H. Vilhjálmsson	Reykjavik University, Iceland
Martin Volk	University of Zurich, Switzerland
Matthew Whelpton	University of Iceland, Iceland
Xiaohong Wu	Minzu University of Qinghai, China

External reviewers

Krasimir Angelov	Marc Dymetman	David Guthrie
Cédric Archambeau	Ramona Enache	Caroline Hagege
Alexandra Balahur	Yang Feng	Guillaume Jacquet
Ruben Izquierdo Bevia	Jochen Frey	Elena Lloret
Leon Derczynski	Bruno Guillaume	Ziad Mikati

Robert Nesselrath
Anna B. Nikulásdóttir
Claude Roux
Felipe Sánchez-Martínez
Gerold Schneider
Gabriel Sekunda
Jinsong Su
Armando Suárez
David Tomás
Zhiyang Wang
Xinyan Xiao
Hao Xiong

Conference Sponsors

POST- AND TELECOM
ADMINISTRATION

Table of Contents

Invited Talks

Research Papers

Reliving the History: The Beginnings of Statistical Machine Translation and Languages with Rich Morphology

Jan Hajič

Institute of Formal and Applied Linguistics, School of Computer Science
Charles University, Prague, Czech Republic
hajic@ufal.mff.cuni.cz

Abstract. In this two-for-one talk, first some difficult issues in morphology of inflective languages will be presented. Then, to lighten up this linguistically and computationally heavy issue, a half-forgotten history of statistical machine translation will be presented and contrasted with current state-of-the art (in a rather non-technical way).

Computational morphology has been on and off the focus of computational linguistics. Only few of us probably remember the times when developing the proper formalisms has been in such a focus; a history poll might still find out that some people remember DATR-II, or other heavy-duty formalisms for dealing with the (virtually finite) world of words and their forms. Even unification formalisms have been called to duty (and the author himself admits to developing one). However, it is not the morphology itself (not even for inflective or agglutinative languages) that is causing the headache – with today's cheap space and power, simply listing all the thinkable forms in an appropriately hashed list is o.k. – but it's the disambiguation problem, which is apparently more difficult for such morphologically rich languages (perhaps surprisingly more for the inflective ones than agglutinative ones) than for the analytical ones. Since Ken Church's PARTS tagger, statistical methods of all sorts have been tried, and the accuracy of taggers for most languages is deemed pretty good today, even though not quite perfect yet.

However, current results of machine translation are even farther from perfect (not just because of morphology, of course). The current revival of machine translation research will no doubt bring more progress. In the talk, I will try to remember the "good old days" of the original statistical machine translation system Candide, which was being developed at IBM Research since the late 80s, and show that as the patents then filed gradually fade and expire, there are several directions, tweaks and twists that have been used then but are largely ignored by the most advanced systems today (including, but not limited to morphology and tagging, noun phrase chunking, word sense disambiguation, named entity recognition, preferred form selection, etc.). I hope that not only this will bring some light to the early developments in the field of SMT and correct some misconceptions about the original IBM system often wrongly labeled as "word-based", but perhaps also inspire new developments in this area for the future – not only from the point of view of morphologically rich languages.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, p. 1, 2010.

Harmonizing WordNet and FrameNet

Christiane D. Fellbaum

Department of Computer Science
Princeton University, Princeton, USA
fellbaum@princeton.edu

Abstract. Lexical semantic resources are a key component of many NLP systems, whose performance continues to be limited by the "lexical bottleneck". Two large hand-constructed resources, WordNet and FrameNet, differ in their theoretical foundations and their approaches to the representation of word meaning. A core question that both resources address is, how can regularities in the lexicon be discovered and encoded in a way that allows both human annotators and machines to better discriminate and interpret word meanings?

WordNet organizes the bulk of the English lexicon into a network (an acyclic graph) of word form-meaning pairs that are interconnected via directed arcs that express paradigmatic semantic relations. This classification largely disregards syntagmatic properties such as argument selection for verbs. However, a comparison with a syntax-based approach like Levin (1993) reveals some overlap as well as systematic divergences that can be straightforwardly ascribed to the different classification principles. FrameNet's units are cognitive schemas (Frames), each characterized by a set of lexemes from different parts of speech with Frame-specific meanings (lexial units) and roles (Frame Elements). FrameNet also encodes cross-frame relations that parallel the relations among WordNet's synsets.

Given the somewhat complementary nature of the two resources, an alignment would have at least the following potential advantages: (1) both sense inventories are checked and corrected where necessary, and (2) FrameNet's coverage (lexical units per Frame) can be increased by taking advantage of WordNet's class-based organization. A number of automatic alignments have been attempted, with variations on a few intuitively plausible algorithms. Often, the result is limited, as implicit assumptions concerning the systematicity of WordNet's encoding or the semantic correspondences across the resources are not fully warranted. Thus, not all members of a synonym set or a subsumption tree are necessarily Frame mates.

We carry out a manual alignment of selected word forms against tokens in the American National Corpus that can serve as a basis for semi-automatic alignment. This work addresses a persistent, unresolved question, namely, to what extent can humans select, and agree on, the context-appropriate meaning of a word with respect to a lexical resource? We discuss representative cases, their challenges and solutions for alignment as well as initial steps for semi-automatic alignment.

(Joint work with Collin Baker and Nancy Ide)

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, p. 2, 2010.

A Morphosyntactic Brill Tagger for Inflectional Languages

Szymon Acedański[1,2]

[1] Institute of Informatics, University of Warsaw,
ul. Banacha 2, 02-097 Warszawa, Poland
accek@mimuw.edu.pl
[2] Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa, Poland

Abstract. In this paper we present and evaluate a Brill morphosyntactic transformation-based tagger adapted for specifics of highly inflectional languages. Multi-phase tagging with grammatical category matching transformations and lexical transformations brings significant accuracy improvements comparing to previous work. Evaluation shows the accuracy of 92.44% for the Polish language which is higher than the same metric for the other known taggers of Polish: stochastic trigram tagger (90.59%) and hybrid tagger TaKIPI employing decision tree classifier and automatically extracted rule-based tagger used for tagging the IPI PAN Corpus of Polish (91.06%).

Keywords: PoS tagger, Brill tagger, inflectional language tagger, morphosyntactic tagger, lexical rules.

1 Introduction

Morphosyntactic tagging is a classic problem in NLP with applications in many higher level processing solutions, namely parsing and then information retrieval, speech recognition and machine translation. Part of Speech tagging for English is already well explored and many taggers have been built with accuracy exceeding 98%. In case of inflectional languages these numbers are much lower, reaching 95,78% for Czech [1] and 92.55% for Polish (per [2]; evaluation by Karwańska and Przepiórkowski [3] reports 91.30%).

The most prominent difference between English and inflectional languages is the size of the tagset. Brill [4] uses Brown's Tagset for English, which consists of almost 200 tags, whereas the IPI PAN Polish tagset [5] contains theoretically over 4000 tags and the manually disambiguated part of the IPI PAN Corpus of Polish [6] used for evaluation contains 1054 different tags. The tags for such languages have a specific structure — along with the part of speech, they contain values of grammatical categories appropriate for the particular part of speech (see Table 1 for an example in Polish). Detailed description of the tagset and the meaning of particular grammatical categories can be found in [5].

Not only the large tagset makes disambiguation a difficult task, but also free word order in considered languages and even problems of unambiguously defining

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 3–14, 2010.

Table 1. Example tags in Brown's English Tagset and IPI PAN Polish tagset

English	`VBD`	verb, past tense
	`PPS`	pronoun, personal, nominative, 3rd person singular
Polish	`praet:sg:m1:perf`	l-participle, singular, human masculine, perfective aspect
	`ppron12:sg:nom:f:pri`	1st person pronoun, singular, nominative, feminine

the correct tags in some cases. Because of this, some corpora allow multiple tags to be assigned to a single segment, whereas other require fully disambiguated tagging, usually providing detailed instructions on how to do this. This matter will be further discussed in the Evaluation section.

Several tagging techniques are commonly known. The most frequently used approaches are: stochastic, e.g., based on Hidden Markov Models [7], and rule-based[1]. Brill [4] presents a transformation-based Part of Speech tagger for English, which automatically chooses good quality transformations given a number of general transformation templates and a training corpus. The tagger used for morphosyntactic disambiguation of the current version of the IPI PAN corpus, called TaKIPI [8], is a hybrid (multiclassifier) transformation-based tagger. Some of the transformations it uses were extracted automatically using machine learning algorithms and then reviewed and adjusted by linguists.

In this paper we describe and evaluate an implementation of the Brill's algorithm, adapted for rich inflectional languages. First steps towards this were described by Acedański and Gołuchowski in 2009 [9], but that tagger was then rewritten with different approaches used in most parts. As in previous work, the adaptation involves splitting the process into phases, so that at first only the part of speech and a few grammatical categories are disambiguated. Remaining categories are determined in the second pass. On top of it, the new, more general approach to transformation templates was developed, and additional transformation templates allowing for transformations which look at particular grammatical categories of surrounding segments were added. Also lexical transformations were used. Finally the tagger was implemented using a new simplified algorithm based on FastTBL [10] and parallelized for better performance.

2 The Original Brill Tagger

Let us describe the original Brill's algorithm in some detail. We assume that we are given three corpora — a large high-quality tagged training corpus, smaller

[1] Throughout this paper, the term *rule-based tagger* is used to denote systems using hand-written rules. For the algorithms involving automatic extraction of rules, the term *transformation-based tagger* is used.

tagged corpus called patch corpus and another one — test corpus, which we want to tag. Brill also assumes, that only one correct tag can be assigned to a segment. Let's denote the tag assigned to i-th segment as t_i.

Tagging is performed in four steps:

1. A simple unigram tagger is trained using the large training corpus.
2. The unigram tagger is used to tag the patch corpus.
3. There are certainly some errors in the tagging of the patch corpus. Therefore we want to generate transformations which will correct as many errors as possible.
   (a) We are given a small list of so called *transformation templates*. Brill uses the following templates in his paper:
      i. $t_i := \mathbf{A}$ if $t_i = \mathbf{B} \ \wedge \ \exists_{o \in \mathbf{O_1}} t_{i+o} = \mathbf{C}$,
      ii. $t_i := \mathbf{A}$ if $t_i = \mathbf{B} \ \wedge \ \forall_{o \in \mathbf{O_2}} t_{i+o} = \mathbf{C_o}$,
      iii. $t_i := \mathbf{A}$ if $t_i = \mathbf{B}$ and i-th word is capitalized,
      iv. $t_i := \mathbf{A}$ if $t_i = \mathbf{B}$ and $(i-1)$-th word is capitalized.

      where
      – $\mathbf{O_1} \in \{\{1\},\{-1\},\{2\},\{-2\},\{1,2\},\{-1,-2\},\{1,2,3\},\{-1,-2,-3\}\}$,
      – $\mathbf{O_2} \in \{\{-2,-1\},\{-1,1\},\{1,2\}\}$,
      – $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{C_o}$ — any tags.
   (b) For each transformation r which can be generated using these templates, we compute two statistics:
      i. $\mathbf{good}(r)$ — the number of places in the patch corpus where the transformation matches and changes an incorrect tag into a correct one,
      ii. $\mathbf{bad}(r)$ — the number of places in the patch corpus where the transformation matches and changes the tagging from correct to incorrect.
   (c) Now we find transformation r_b, which maximizes $\mathbf{good}(r) - \mathbf{bad}(r)$, i.e. reduces the largest possible number of errors when applied. We save the transformation and apply it to the patch corpus. If the patch corpus still contains many errors, return to 3a.
4. The test corpus is first tagged using the unigram tagger, and then the saved transformations are applied in order.

If the test corpus was previously manually tagged, we can evaluate the performance of the tagger.

3 Adaptation for Inflectional Languages

The algorithm described in the previous section was subsequently extended by applying a number of techniques targeted at improving accuracy of tagging of inflectional languages. These techniques are:

– multi-pass tagging — gradually disambiguating parts of tags,
– generalized transformation templates — allowing for more flexible design and then specific templates for inflectional languages relying on interdependencies between values of grammatical categories,

– lexical transformation templates — allowing to match prefixes and suffixes of processed lexemes,
– simplified implementation of FastTBL algorithm [10] and parallelization of the tagging engine for good performance and maintainability

3.1 Multi-pass Tagging

The first technique is used to reduce the size of the transformation space and to avoid too specific transformations in the first stage. It is inspired by [9], where the authors split the tags into two parts sharing only the part-of-speech. In the first run of the Brill tagger the tagset consists of only one of the parts of tags. In the second run, the tagset comprised of the other parts is used, but the previously selected parts of speech are fixed for the second pass. Also Tufiş [11] proposes using a reduced tagset with easily recoverable grammatical categories not present, to improve performance. Out goal is different though — we try to leave some of the hard to disambiguate categories for later stages so that the tagger already has more information from preceding phases.

We consider a sequence T_i ($i \in \{0, \ldots, k-1\}$) of gradually simplified tagsets. T_0 is the original tagset and T_{j+1} ($j \in \{0, \ldots, k-2\}$) are some other tagsets. Projections mapping specific tags to more general tags are also needed: $\pi_j \colon T_j \to T_{j+1}$. For each of the tagsets a separate pass of the Brill algorithm is performed. The tag assigned to the i-th segment in the p-th pass ($p \in \{1, \ldots, k\}$) is denoted by t_i^p. In the first pass the simplest tagset T_{k-1} is used. In the p-th pass, for i-th segment, only tags $t_i^p \in T_{k-p}$ are considered such that $\pi_{k-p}(t_i^p) = t_i^{p-1}$.

In our experiments we used only two tagsets — T_0 being the original and T_1 which had information about part of speech, case and person only. π_0 was a natural projection i.e. the one which strips values of grammatical categories not present in T_1. The produced software can be configured for more than two phases with different tagsets.

3.2 Generalized Transformation Templates

In the original Brill classifier, all the transformation templates are of the following form:

> Change t_i to **A** if it is **B** and ....

In our tagger we generalize the possible transformation templates by allowing other operations than changing the entire tag to be performed. Also, the current tag of a lexeme need not be fully specified in an instantiation of some transformation template.

A particular transformation template consists of a *predicate* template which specifies conditions on the context where the transformation should be applied, and an *action* template describing the operation to be performed if the predicate matches. For example in the transformation template "change the tag to **A** if the tag is **B**", the first part ("change the tag to **A**") is the action template and the

second part ("the tag is **B**") is the predicate template. The same nomenclature is applied to instantiated transformations.

This generalization was performed in order to allow using more general transformations than allowed by the original algorithm. Let's denote by $t_i|_{\text{CASE}}$ the value of grammatical category CASE in the tag of the i-th segment of the text. Now consider the very robust linguistic rule "if an adjective is followed by a noun, then they should agree in CASE".
This rule may be composed of

- an *action*: $t_i|_{\text{CASE}} := t_{i-1}|_{\text{CASE}}$
- a *predicate*: $t_i|_{\text{POS}} = \text{SUBST} \ \wedge \ t_{i-1}|_{\text{POS}} = \text{ADJ}$

The proposed tagger is able to generate transformations resembling such rules. It uses the following predicate templates:

1. $t_i^p = \mathbf{T} \ \wedge \ \exists_{o \in \mathbf{O}} t_{i+o}^p = \mathbf{U}$,
2. $t_i^p = \mathbf{T} \ \wedge \ \forall_{o \in \mathbf{O}} t_{i+o}^p = \mathbf{U_o}$,
3. $t_i^p = \mathbf{T} \ \wedge \ \exists_{o \in \mathbf{O_1}} t_{i+o}^{p-1} = \mathbf{U'}$,
4. $t_i^p|_{\text{POS}} = \mathbf{P} \ \wedge \ t_i^p|_{\mathbf{C}} = \mathbf{X} \quad \wedge \quad \exists_{o \in \mathbf{O}} \left(t_{i+o}^p|_{\text{POS}} = \mathbf{Q} \ \wedge \ t_{i+o}^p|_{\mathbf{C}} = \mathbf{Y} \right)$,
5. $t_i^p|_{\text{POS}} = \mathbf{P} \ \wedge \ t_i^p|_{\mathbf{C}} = \mathbf{X} \quad \wedge \quad \forall_{o \in \mathbf{O}} \left(t_{i+o}^p|_{\text{POS}} = \mathbf{Q_o} \ \wedge \ t_{i+o}^p|_{\mathbf{C}} = \mathbf{Y_o} \right)$,

and action templates:

1. $t_i^p := \mathbf{V}$,
2. $t_i^p|_{\text{POS}} := \mathbf{R}$,
3. $t_i^p|_{\mathbf{C}} := \mathbf{Z}$,

where

- $\mathbf{T}$, $\mathbf{U}$, $\mathbf{U_o}$, $\mathbf{V}$ — any tags valid in pass p,
- $\mathbf{U'}$ — any tag valid in pass $p-1$,
- $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{Q_o}$, $\mathbf{R}$ — any parts of speech valid in pass p,
- $\mathbf{C}$ — any grammatical category valid in pass p,
- $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Y_o}$, $\mathbf{Z}$ — any values valid for category $\mathbf{C}$,
- $\mathbf{O} \in \{\{1\}, \{-1\}, \{2\}, \{-2\}, \{1,2\}, \{-1,-2\}, \{1,2,3\}, \{-1,-2,-3\}\}$,
- template variables $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{Q_o}$ (for all o at the same time), $\mathbf{R}$, $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Y_o}$ (for all o at the same time) and $\mathbf{Z}$ could have a special value $\star$ meaning *any*.

Additionally, the actions were implemented in such a way, that they were not applied if they were to assign a tag not reported by the morphological analyzer for a particular segment. In case of actions 2 and 3, the nearest possible tag was used instead. The metric used here is the number of matching values of grammatical categories, but only tags with the expected part of speech are considered. If no such tags are possible, the action is not performed.

3.3 Lexical Transformations

Another extension which proved very useful are lexical transformation templates proposed by Brill in a later paper [12]. Megyesi [13] subsequently explored them for Hungarian (which is an agglutinative language, with a number of affixes possessing grammatical functions). The results were very promising. The author used the following predicate templates:

1. $t_i^p = \mathbf{T} \ \wedge \ orth_i$ contains letter $\mathbf{L}$,
2. $t_i^p = \mathbf{T} \ \wedge \ orth_i$ starts/ends with $\mathbf{S}$, $|\mathbf{S}| < 7$,
3. $t_i^p = \mathbf{T} \ \wedge \ orth_i$ with deleted prefix/suffix $\mathbf{S}$, $|\mathbf{S}| < 7$, is a word,
4. $t_i^p = \mathbf{T} \ \wedge \ (orth_{i_1} = \mathbf{W} \vee orth_{i+1} = \mathbf{W})$.

Here $orth_i$ simply denotes the orthographic representation of the i-th segment. Inspired by this work, and after some experiments, we extended the list of predicate templates by only the prefix/suffix matching:

1a. $t_i^p = \mathbf{T} \ \wedge \ orth_i$ ends with $\mathbf{S}'$

1b. $t_i^p = \mathbf{T} \ \wedge \ orth_i$ starts with $\mathbf{S}'$

4a. $t_i^p|_{\mathrm{POS}} = \mathbf{P} \ \wedge \ t_i^p|_{\mathbf{C}} = \mathbf{X} \ \wedge \ orth_i$ ends with $\mathbf{S}$
$\wedge \ \exists_{o \in \mathbf{O}} \left(t_{i+o}^p|_{\mathrm{POS}} = \mathbf{Q} \ \wedge \ t_{i+o}^p|_{\mathbf{C}} = \mathbf{Y} \right)$

4b. $t_i^p|_{\mathrm{POS}} = \mathbf{P} \ \wedge \ t_i^p|_{\mathbf{C}} = \mathbf{X}$
$\wedge \ \exists_{o \in \mathbf{O}} \left(t_{i+o}^p|_{\mathrm{POS}} = \mathbf{Q} \ \wedge t_{i+o}^p|_{\mathbf{C}} = \mathbf{Y} \right.$
$\left. \wedge \ orth_{i+o} \text{ ends with } \mathbf{S} \right)$

where $\mathbf{S}$ and $\mathbf{S}'$ are any strings no longer than 3 and 2 characters respectively. This resulted in over 1.5% accuracy improvement over the Brill tagger with only generalized transformations, as tested for Polish.

3.4 Simplified FastTBL Implementation

The idea behind the FastTBL algorithm [10] is the minimization of the number of accesses to data structures for storing $good(\cdot)$ and $bad(\cdot)$ functions. Unfortunately this comes with the increased complexity — there are 8 possible branches of execution in the main loop. Therefore we designed a simplified version of this algorithm presented as Algorithm 1. It allows redundant updates of $good(\cdot)$ and $bad(\cdot)$, but this extra work does not significantly influence the total running time of the algorithm, because the most computationally intensive work is generating the possible transformations in lines 2, 20 and 27, as well as the application of the generated transformation in 18.

3.5 Parallelization

Finally, the tagger was implemented specifically for multiprocessing environment, mostly because of the high memory requirements for storing the $good(\cdot)$ and $bad(\cdot)$ functions. The ordinary 32-bit Linux operating system originally used for experiments does not allow for more than 2GB of memory per process. The OpenMPI [14] library was used, which also gives the possibility to run the tagger on multiple machines in parallel in a standardized way.

The workload is split between processes by distributing the transformation templates considered above (and the values of the $\mathbf{O}$ template variable, see section 3.2) among them. Therefore every process stores only a part of all the transformations. In each round of the algorithm the best transformation is collectively found, broadcast to all the processes, and the processing continues as shown in Algorithm 1.

Algorithm 1. Pseudocode of the simplified FastTBL algorithm

```
1. {Initializing good and bad data structures}
2. for i = 0 to len(text) do
3.    for each transformation r which matches at position i do
4.       if r corrects the classification of i-th segment then
5.          increase good(r)
6.       else if r changes the classification of i-th segment from correct to wrong then
7.          increase bad(r)
8.       end if
9.    end for
10. end for
11.
12. {Main loop — generating transformations}
13. loop
14.    b := arg max_r(good(r) − bad(r)) {the best transformation}
15.    if good(b) − bad(b) < threshold then
16.       return
17.    end if
18.    add b to the sequence of generated transformations
19.    text' := text after application of b
20.    for each position i in vicinity of the changes performed by b do
21.       for each transformation r which matches at position i in text do
22.          if r corrected the classification of i-th segment in text then
23.             decrease good(r)
24.          else if r miscorrected the classification of i-th segment in text then
25.             decrease bad(r)
26.          end if
27.       end for
28.       for each transformation r which matches at position i in text' do
29.          if r corrects the classification of i-th segment in text' then
30.             increase good(r)
31.          else if r miscorrects the classification of i-th segment in text' then
32.             increase bad(r)
33.          end if
34.       end for
35.    end for
36.    text := text'
37. end loop
```

4 Evaluation

The tagger was evaluated on two corpora of Polish: the IPI PAN Corpus of Polish [6] and the new National Corpus of Polish [15] (in preparation; version dated 2009–12–16 was used). The former corpus is allowed to have multiple golden tags for one segment, whereas the latter is fully disambiguated. For evaluation the manually disambiguated subcorpora were used, of size 880 000 and 648 000 segments, respectively.

The methodology proposed in [3] was used (which was also employed in [16]). A corpus was split into training part and evaluation part by ratio 9:1. The training part was used both as the training and patch corpus of the Brill algorithm. All taggers were configured to choose exactly one tag for each segment. Ten-fold cross-validated results are presented in Tables 2 and 3.

Table 2. Evaluation results — IPI PAN Corpus. Sources: [3,16]

Tagger	Full tags					PoS only				
	C	*WC*	*P*	*R*	*F*	*C*	*WC*	*P*	*R*	*F*
Trigram HMM [17]	87.39	90.59	84.51	83.09	83.80	96.79	97.11	96.75	96.78	96.77
TaKIPI [2]	88.68	91.06	90.94	83.78	87.21	96.53	96.54	96.58	96.71	96.65
Brill [9] (2009)			89.46	83.55	86.40					
Brill (this paper)	90.00	92.44	92.44	86.05	89.13	98.17	98.18	98.18	98.16	98.17

Table 3. Evaluation results — National Corpus of Polish (full tags)

Thr[a]	Time (s)	# transformations		*Acc.* (%)
		P1	P2	
2	1450	5175.1	2748.0	92.82%
6	632	1422.6	612.2	92.68%

[a] The minimum $good(r) - bad(r)$ of the generated transformations.

Correctness (C) — percent of segments for which the tagger assigned exactly the same set of tags as the golden standard. Please note that in the IPI PAN corpus for some segments several tags are marked correct.

Weak correctness (WC) — percent of segments for which the sets of interpretations determined by the tagger and the golden standard are not disjoint.

Precision (P) — percent of tags (given by the morphological analyzer) which were both chosen by the tagger and the golden standard.

Recall (R) — percent of golden tags which were chosen by the tagger.

F-measure (F) — $F = \frac{2PR}{P+R}$.

Accuracy (*Acc.*) — any of the above in the case of the National Corpus of Polish, which has always one golden tag per segment.

The times in Table 3 were obtained on a multiprocessor machine with Xeon CPUs clocked at 2.4 GHz (with 12MB cache). 6 processes were run. The tagger was compiled in 64-bit mode, which probably negatively impacted performance due to almost doubled memory usage (∼1.2 GB per process compared to ∼0.7 GB in similar 32-bit setup), but this was not verified.

It is also worth noting that only TaKIPI does not disambiguate words not known to the morphological analyser, even if the input contains a number of possible morphosyntactic interpretations.

To provide a better insight into the classes of errors generated by the tagger, detailed statistics are presented in Tables 4, 5 and 6. It can be clearly seen that the most problems the tagger has concern CASE and GENDER. Slightly fewer errors are reported for NUMBER. This is similar to previous findings and not unexpected for Polish. Nevertheless, the introduction of lexical elements in transformation templates gave over 1.5% improvement in accuracy (on the National Corpus of Polish). Over 60% of all generated transformations do contain lexical matchers. The vast majority of them is used for determining the correct CASE by matching nearby segments' suffixes (see Table 7). Also, they are used for disambiguating rare flexemic classes like QUB from CONJ.

There are also some categories of errors in the testing corpush, which would not be disambiguated by a human looking at the same amount of context. Let us present several examples:

— Long nominal phrases, especially near sentence or subordinate clause boundaries:

Table 4. Error rates for parts of speech (shown only values > 0.01%)

Expected PoS	# errs	% toks
subst	3028	0.47%
qub	1658	0.26%
adj	1596	0.25%
ger	1392	0.21%
conj	652	0.10%
adv	597	0.09%
ppas	522	0.08%

Expected PoS	# errs	% toks
prep	471	0.07%
pred	363	0.06%
num	326	0.05%
fin	268	0.04%
pact	208	0.03%
comp	172	0.03%

Table 5. Error rates for grammatical categories (shown only values > 0.01%)

Category	# errs	% toks
CASE	21259	3.28%
GENDER	16151	2.49%
NUMBER	4645	0.72%
ASPECT	416	0.06%
ACCOMMODABILITY	193	0.03%

Table 6. Specific errors in assignment of grammatical categories (top 15 records)

Expected	Actual	# errs	% toks
CASE(NOM)	CASE(ACC)	7188	1.11%
CASE(ACC)	CASE(NOM)	4717	0.73%
GENDER(M1)	GENDER(M3)	2543	0.39%
CASE(GEN)	CASE(ACC)	2533	0.39%
NUMBER(SG)	NUMBER(PL)	2460	0.38%
NUMBER(PL)	NUMBER(SG)	2185	0.34%
GENDER(M3)	GENDER(M1)	1989	0.31%
GENDER(M3)	GENDER(N)	1662	0.26%
GENDER(M1)	GENDER(F)	1375	0.21%
GENDER(F)	GENDER(M3)	1243	0.19%
GENDER(M3)	GENDER(F)	1214	0.19%
GENDER(F)	GENDER(N)	1115	0.17%
CASE(GEN)	CASE(NOM)	1105	0.17%
CASE(ACC)	CASE(GEN)	963	0.15%
GENDER(N)	GENDER(M3)	907	0.14%

Table 7. Sample lexical transformations generated by the tagger in the first pass

No.	r	**good**(r)	**bad**(r)
3	Change CASE of preposition from ACC to LOC if it ends with `na` (in practice this asks for the particular preposition *na*, in English: *on*) and one of two following segments has CASE of LOC.	2496	113
7	Change CASE of an adjective from LOC to INST if one of the three following segments has CASE of INST and ends with `em`.	921	29

```
... tego znaku. Zamiłowanie do sportu i ...
... this sign. Passion for sport and ...
```

Here the underlined word can have either nominal or accusative case.

— Expressions with words like *państwo*, which may be either a noun (*country*) or a pronoun (formal plural *you*):

```
..., o czym państwo w tej chwili ...
..., what you/the country at the moment ...
```

This calls for enlarging the lookup context in the future. For example, predicates like "the nearest segment with part-of-speech **P** has category **C** equal **X**" may be good candidates for inclusion. This requires extending the vicinity parameter, and therefore slows down the computations, but may result in better accuracy.

5 Conclusions and Future Work

The paper presents and evaluates a number of techniques designed to adapt Brill tagger for inflectional languages with large tagsets. Especially adding predicates

and actions which allow matching or changing values of single grammatical categories, as well as adding lexical transformations, were the most valuable modifications of the original algorithm.

It is worth noting that the tagger does not need any linguistic knowledge provided, except the specification of tagsets and the information about the grammatical categories which should be disambiguated in consecutive phases. Rule templates are not designed for any specific language. Even if some transformation templates are not suitable for considered language, they may negatively impact only performance, but not accuracy.

As far as the quality of the new tagger is concerned, the reported numbers are at least 1.1% higher than for other existing taggers for Polish, although this should be independently verified. Also, it may be an interesting experiment to use the tagger for other languages, like Czech or Hungarian (maybe after inclusion of all lexical transformations proposed by Megyesi [13]). There are also some other places for improvement not explored yet, namely:

1. Experimenting with different simplified tagsets and more than 2 passes. Tufiş [11] proposes using an additional reduced tagset to collapse grammatical categories which are unambiguously recoverable from the lexicon. This reduces the transformation space, improving performance. Others suggest joining some parts of speech or values of grammatical categories which have similar grammatical functions in the first pass, to disambiguate them later. For example in an intermediate phase one would use the value NOM-OR-ACC for CASE,
2. Simply enlarging the context of transformation templates may be a good way to go,
3. Designing transformation templates which look for the nearest segment with a particular part of speech or value of some grammatical category may improve accuracy.

The full source code of the tagger is available under the terms of the GNU GPL v3 from its project page: `http://code.google.com/p/pantera-tagger/`.

Acknowledgments. I'd like to sincerely thank my academic advisor Prof. Adam Przepiórkowski for his valuable help, always inspiring talks and effective motivation.

The research is funded in 2007–2010 by a research and development grant from the Polish Ministry of Science and Higher Education.

References

1. Spoustová, D.: Combining Statistical and Rule-Based Approaches to Morphological Tagging of Czech Texts. The Prague Bulletin of Mathematical Linguistics (89), 23–40 (2008)
2. Piasecki, M., Godlewski, G.: Effective Architecture of the Polish Tagger. In: Sojka, P., Kopecek, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 213–220. Springer, Heidelberg (2006)

3. Karwańska, D., Przepiórkowski, A.: On the Evaluation of Two Polish Taggers. In: Proceedings of the 2009 PALC Conference in Łódź, Frankfurt/M., Peter Lang (2009) (to appear)
4. Brill, E.: A simple rule-based part of speech tagger. In: Proceedings of the Third Conference on Applied Natural Language Processing, Morristown, NJ, USA, pp. 152–155. Association for Computational Linguistics (1992)
5. Przepiórkowski, A., Woliński, M.: A Flexemic Tagset for Polish. In: Proceedings of Morphological Processing of Slavic Languages, EACL 2003 (2003)
6. Przepiórkowski, A.: The IPI PAN Corpus: Preliminary version. Institute of Computer Science, Polish Academy of Sciences, Warsaw (2004)
7. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing. In: Computational Linguistics and Speech Recognition, February 2008, 2nd edn. Prentice Hall, Englewood Cliffs (2008)
8. Piasecki, M., Wardyński, A.: Multiclassifier Approach to Tagging of Polish. In: Proceedings of 1st International Symposium Advances in Artificial Intelligence and Applications, unknown (2006)
9. Acedański, S., Gołuchowski, K.: A Morphosyntactic Rule-Based Brill Tagger for Polish. In: Recent Advances in Intelligent Information Systems, Kraków, Poland, June 2009, pp. 67–76. Academic Publishing House EXIT (2009)
10. Ngai, G., Florian, R.: Transformation-based learning in the fast lane. In: NAACL 2001 on Language technologies, Morristown, NJ, USA, pp. 1–8. Association for Computational Linguistics (2001)
11. Tufis, D.: Tiered Tagging and Combined Language Models Classifiers. In: Matoušek, V., Mautner, P., Ocelíková, J., Sojka, P. (eds.) TSD 1999. LNCS (LNAI), vol. 1692, pp. 28–33. Springer, Heidelberg (1999)
12. Brill, E.: Some advances in transformation-based part of speech tagging. In: AAAI 1994: Proceedings of the Twelfth National Conference on Artificial Intelligence, Menlo Park, CA, USA, vol. 1, pp. 722–727. American Association for Artificial Intelligence, Standford (1994)
13. Megyesi, B.: Improving Brill's POS tagger for an agglutinative language. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 275–284 (1999)
14. Gabriel, E., et al.: Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. In: Kranzlmüller, D., Kacsuk, P., Dongarra, J. (eds.) EuroPVM/MPI 2004. LNCS, vol. 3241, pp. 97–104. Springer, Heidelberg (2004)
15. Przepiówski, A., Górski, R.L., Lewandowska-Tomaszyk, B., Łaziński, M.: Towards the National Corpus of Polish. In: Proceedings of the Sixth International Language Resources and Evaluation (2008)
16. Acedański, S., Przepiórkowski, A.: Towards the Adequate Evaluation of Morphosyntactic Taggers. In: Proceedings of the 23rd International Conference on Computational Linguistics (2010) (to appear)
17. Dębowski, Ł.: Trigram morphosyntactic tagger for Polish. In: Intelligent Information Systems. Advances in Soft Computing, pp. 409–413. Springer, Heidelberg (2004)

Hybrid Syntactic-Semantic Reranking for Parsing Results of ECAs Interactions Using CRFs

Enzo Acerbi[1], Guillermo Pérez[1], and Fabio Stella[2]

[1] Julietta Research Group, University of Seville
{enzoace,gperez}@us.es
[2] University of Milano-Bicocca
stella@disco.unimib.it

Abstract. Reranking modules of conventional parsers make use of either probabilistic weights linked to the production rules or just hand crafted rules to choose the best possible parse. Other proposals make use of the topology of the parse trees and lexical features to reorder the parsing results. In this work, a new reranking approach is presented. There are two main novelties introduced in this paper: firstly, a new discriminative reranking method of parsing results has been applied using Conditional Random Fields (CRFs) for sequence tagging. Secondly, a mixture of syntactic and semantic features, specifically designed for Embodied Conversational Agents (ECAs) interactions, has been used. This approach has been trained with a Corpus of over 4,000 dialogues, obtained from real interactions of real users with an online ECA. Results show that this approach provides a significant improvement over the parsing results of out-of-domain sentences; that is, sentences for which there is no optimal parse among the candidates given by the baseline parse.

Keywords: Embodied conversational agents, natural language processing, dialogue systems, sequence tagging, CRFs.

1 Introduction

1.1 Embodied Conversational Agents

Conversational Agents (CAs) can be defined as "*communication technologies that integrate computational linguistics techniques with the communication channel of the Web to interpret and respond to statements made by users in ordinary natural language*" [1]. Embodied Conversational Agents (ECAs) are empowered with a human representation that shows some degree of empathy (smiling, showing sadness, disgust) with the user as the dialogue goes on.

The fact of adding explicit anthropomorphism in Conversational Agents has some effects over the solution designed:

– A number of the user interactions are actually social dialogue or "small-talk", where the users interact with the ECA informally [2]

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 15–26, 2010.

– Users may perceive the combination of embodied characters with advanced natural language processing techniques and social dialogue strategies positively. But on the other hand, if the language understanding performance or the social dialogue strategies behave poorly, users perceive the solution worse than the equivalent text-only chatbot without any character [3], [4].

Natural language processing for commercial ECAs applications shows some peculiarities. Usually, customers and service providers come to an agreement on the set of questions and services that the final users may request to the ECA. Customers demand optimal performance and fast reaction time over the previously agreed domain. This implies that these in-domain utterances from the user have to be accurately parsed, while some degree of flexibility can be tolerated in the rest of the sentences. A common approach to cope with these requirements is to divide the lexical items into two groups: those that belong to the agreed Corpus and the rest of the words. The first group is configured using domain specific semantic labels while the second one is assigned common syntactic categories.

Similarly, the production rules at grammar level are semantically oriented for the sentences included in the ECA's Corpus and syntactically oriented for utterances that don't belong to the ECA's Corpus.

This work has been trained over a set of 4,000 sentences from real users to an online ECA. The application domain is a Corpus of common questions asked to the customer service of a furniture retail company. Examples of these questions are:

1. What are your opening hours?
2. How much does a sofa cost?

Along with the retail specific questions, there is a wide coverage of general questions included. These questions include flirting interactions, insults, compliments and general knowledge (politics, sport, etc.). This coverage is treated as part of the domain configuration and is known as the "social configuration" or "personality" of the ECA.

1.2 Related Work

The idea of discriminative reranking of parsing results is not new. In [5], [6] the authors propose a reranking process over the parsing results using a Maximum Entropy approach. Also Collins [7] propose a similar strategy making use of Markov Random Fields and boosting approaches, achieving significant improvement on error rate over the baseline system performance.

The approaches detailed in those papers are based on lexical and syntactic features describing the components of the parse tree and their syntactic relationship. The reranking layer is applied over a set of candidates which are obtained with a classical generative parser.

In [8] an application of the previous proposals for semantic parsing is described. In addition to the purely syntactic features, the authors include semantic features on the reranking process, obtaining partial improvements.

In this paper radical a different strategy is proposed: all parse tree structure is ignored and only terminal symbols are taken into account. To our knowledge, there is no previous work on reranking parsing results making use of sequence labeling as reranking method.

1.3 Generative Parser

The approach hereby described relies on a set of candidate parsing results provided by a generative parser. The parser used in the experiments was [9], [10], a unification grammar based context free parser inspired in the Lexical Functional Grammar formalism [11]. The parsing results are therefore provided by means of two different structures: the F-structure and the C-structure. The first one is a set of language independent attribute–value pairs while the second one is the language–dependent parse tree.

Regarding the parsing strategy, the previously described mixture syntactico-semantic approach has been followed: semantically oriented lexical and grammatical description for the domain and personality Corpora, and a syntactically oriented configuration for the other utterances. When the parser provides a parse with plain syntactic labels of an incoming sentence, the ECA uses it to look up the customer's web site for pages where the mentioned terms are included. On the other hand, when the parser provides a parse with semantic labels, the ECA returns the appropriate preconfigured answer with that representation or engages in a subdialogue with the user.

The baseline system make use of a set of heuristic domain-independent rules to choose the best candidate. These rules take into account the tree structure of the parsing results. Some of the rules are specifically designed for ECAs interactions.

2 Conditional Random Fields

CRFs are probabilistic undirected graphical models. Each vertex of the CRF represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. $\mathbf{X}$ is the sequence of observations and $\mathbf{Y}$ is the sequence of unknown state variables that needs to be inferred based on the observations. In the application hereby described, $\mathbf{X}$ is formed by all the words of the sentence, while $\mathbf{Y}$ is the sequence of their corresponding labels. CRFs are especially suitable for sequence labeling problems since independence assumptions are made among $\mathbf{Y}$ but not among $\mathbf{X}$. Thats is, CRFs allow the use of strong interdependent and overlapping features, as required by sequence labeling problems.

Formally, CRFs can be defined as follows [12]: *Let $G = (V, E)$ be a graph such that $\boldsymbol{Y} = (\boldsymbol{Y}_v)_{v \in V}$, so that $\boldsymbol{Y}$ indexed by the vertices of G. Then $(\boldsymbol{X}, \boldsymbol{Y})$ is a conditional random field in case, when conditioned on $\boldsymbol{X}$, the random variables $\boldsymbol{Y}_v$ obey the Markov property with respect to the graph: $p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \neq v) = p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G.* The likelihood probability for the CRF model θ is calculated in this way: $p_\theta(\mathbf{y} \mid \mathbf{x}) =$

$$exp\left\{\sum_{e\in E,k}\lambda_k f_k(e,\mathbf{y}\mid_e,\mathbf{x})+\sum_{v\in V,k}\mu_k g_k(v,\mathbf{y}\mid_v,\mathbf{x})\right\} \quad (1)$$

Notice that λ and μ represent the model's parameters and f and g represent the feature functions that are provided to the model in the training phase.

3 A New Approach

3.1 Parse Trees as Lexical Sequences

A key observation that allows this new approach is the fact that no pair of alternative trees provided by the generative parser share the same sequence of lexical categories. This statement is true because the syntactic ambiguity is locally solved by the baseline parser before providing the alternative trees to the reranking module. In other words, to distinguish one parse tree from another, one can just look at the categories assigned to each word in the sentence. Therefore, the problem of finding the optimal parse boils down to finding the optimal assignment of the lexical category for each word in the sentence, among those given by the parser. Thus, a new parse tree sequence representation is proposed. The problem of reranking parsing results is therefore reduced to a word-category assignment: the new problem is to find the best assignment for the whole sentence, which is a typical sequence labeling problem.

The sequence labeling problem is faced with up to 223 different labels:

- 13 classical syntactic lables (noun, verb, etc.)
- 210 domain specific semantic labels (furniture, price, etc.)
- 2 additional labels:
  - One to describe the lexical items not included in the best alternative.
  - One to identify the lexical items not included in the best alternative abut located in the middle of two partial sequences.

3.2 New Problem Characteristics

The reranking approach described in this paper is conditioned by the following issues:

- The parser handles a mixture of syntactic and semantic lexical categories and grammar production rules, with overlapping syntactic-semantic alternative trees.
- The reranking algorithm must face an elevated tagset dimension with 223 different labels. High dimensional tagsets like this one could make the problem intractable.

4 The Proposed Solution

4.1 Theory

The strategy to keep the problem tractable despite the tagset dimension is based on helping the model in two major ways. The first one is through the introduction of highly informative features in order to reduce the tagset dimension for every specific word. This goal is achieved by exploiting a priori knowledge about a term. Secondly, the model prediction is driven; the model is not asked to directly predict the correct label sequence: instead, the likelihood of every sequence is used for optimal selection. Additionally, the training set size is high enough to ensure the presence of "past cases" for every label in the tagset.

Since words in a sentence are strongly interdependent, the solution has to be able to model dependencies between entities; moreover, words can be linked to a big set of features that can help classification, but dependencies may exist also between features.

One of the most well-known approaches to sequence labeling is Hidden Markov Models [13]. The potential problem using HMMs is that they calculate $p(x \mid y)$, where x is the word, and y is the label. The point is that what really needs to be modeled is $p(y \mid x)$. A solution can be Maximum Entropy Markov Models (MEMM), where $p(y \mid x)$ is calculated using a maximum entropy model. But MEMM can suffer the label bias problem.

CRFs are a suitable model for the task at hands, since they do not suffer the label bias problem; they are not per-state normalized like MEMMs: instead of being trained to predict each label independently, they are trained to get the whole sequence correctly.

4.2 Implementation

Offline. Due to the specificity of the problem, the creation of an *ad hoc* training set has been necessary in order to take into account domain-dependent semantic categories. The training set is formed by a Corpus of over 4,000 dialogues of Human-ECAs interactions and all the alternative parse trees for every sentence.

During the offline phase, the correct alternative has been manually tagged for every sentence. The tagging process was done by choosing among a set of sequence-like representation of the parse trees. The tagger application graphically shows, for each sentence, all the possible sequences of lexical categories and allows to select the best sequence or the best combination of sequences. Figure 2 shows a screen-shot of the application.

If the sequence selected does not include all the words in the sentence, the excluded words are labelled as *Not_Used*. Sometimes the correct parse tree of a sentence is captured by a combination of two or more partial sequences. In order to prevent the bad tendency of the model to predict too many words as *Not_Used*, words between two partial sequences are classified as *Link*. Thus the model learns to distinguish between words that can be ignored, namely *Not_Used*, and words that are functioning as a bridge between partial sequences, namely *Link*. Figure 3 shows the merging of the two partial sequences selected in Figure 2.

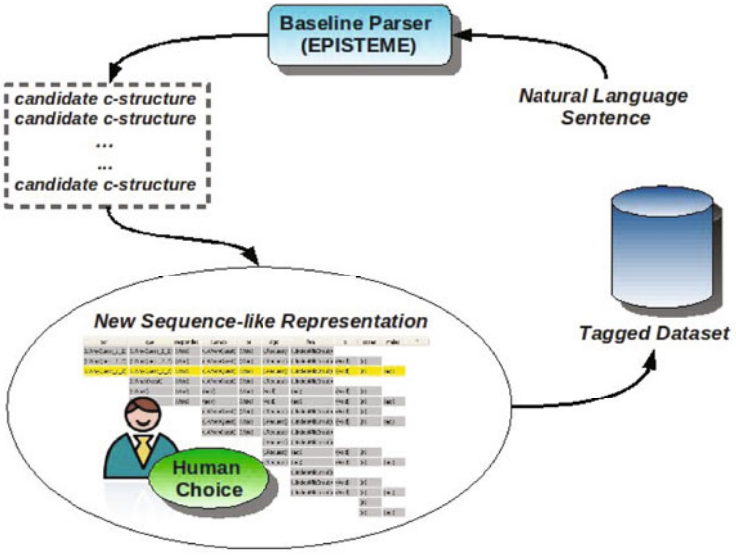


Fig. 1. Offline processing of the proposed approach

Sentences can be classified in two main categories:

- **In-domain sentences:** Sentences for which an optimal full-parse sequence or an optimal combination of partial sequences can be obtained among the candidates given by the baseline parser.
- **Out-of-domain sentences:** Sentences for which an optimal full-parse sequence or an optimal combination of partial sequences can not be obtained among the candidates given by the baseline parser.

Both kinds of sentences were tagged, but only in-domain sentences were used to train the model.

The following table provides some data about the distribution of in-domain and out-of-domain sentences in the dataset:

	Number of Sentences	Number of Words	Average length
In-domain	4,096	32,134	8.2
Out-of-domain	1,011	15,712	13.8

Besides these two groups, extremely bad formed sentences were classified as *No Parse* and discarded in the training phase (5% of the total amount of analyzed sentences).

The tagging application allows the user to choose the correct sequence in a time between 5 and 15 seconds approximately, depending on the sentence complexity. The final average tagging rate was 100 sentences per hour; the entire tagging process took over 50 hours.

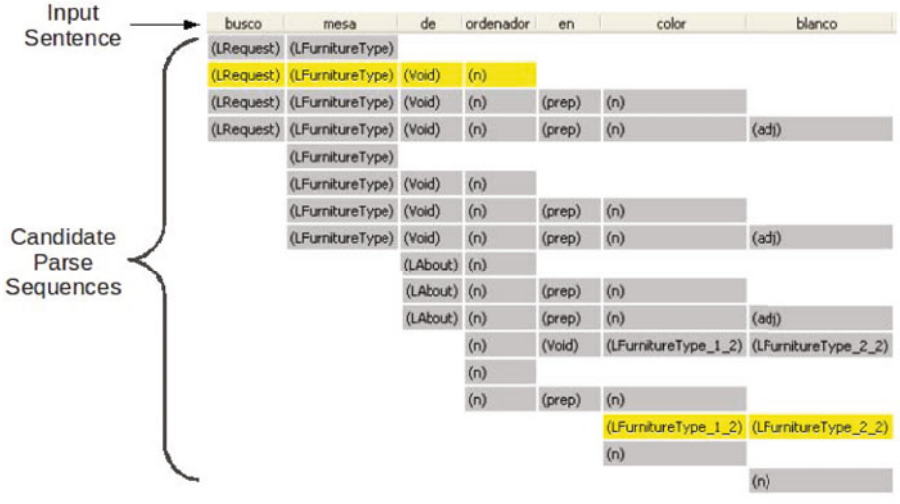


Fig. 2. A detail of the application created for corpus tagging

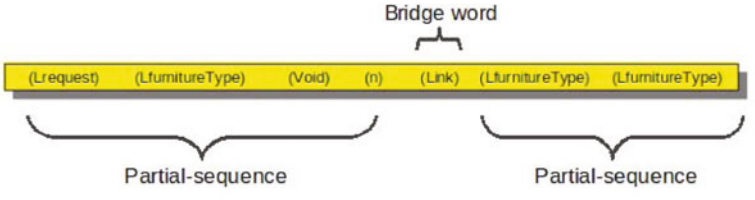


Fig. 3. Merging sequences in a unique global sequence

The MALLET library [14] was used for building the CRF model and has been modified to obtain the likelihood associated with each candidate. The model was validated with a 5-fold cross validation; details about the dataset are provided in the Experimental Results section.

Online. The online phase refers to the real interactions between the ECA and a user. Figure 4 shows the way the CRF model is used at running time: the natural language sentence provided by the user is analyzed by the baseline parser and a set of candidate sequences is returned. At this point, all possible combinations of partial sequences have to be generated and added to the original set of candidates. The CRFs model trained in the offline phase returns the likelihood probability associated with each candidate sequences. The highest likelihood sequence is identified as the optimal one, and the related c-structures (one or more) chosen.

Features. As previously mentioned, a set of highly informative features was introduced in order to limit the number of possible labels for a specific word.

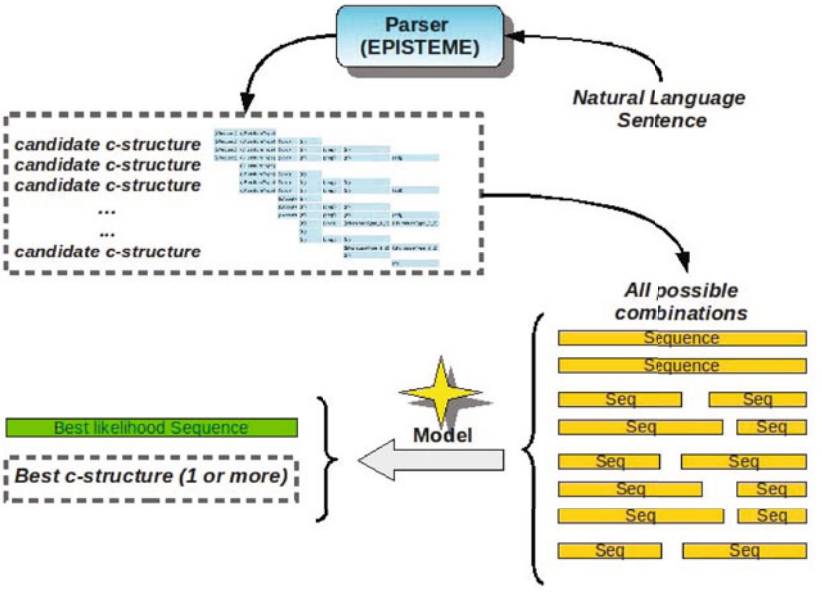


Fig. 4. Online processing

If it is known that word x can be classified only as tag_1, tag_2, tag_3 and tag_4 and this information is properly introduced into the model, it allows the model to focus the prediction on the specific subset, ignoring remaining label assignments.

Three kind of features were used:

- The root or lemma of the word.
- Word related features: the highly informative features described above. They consist of a large set of binary features indicating if the word belongs or not to a specific subdictionary. For example, if the word *beautiful* appears in the *nouns*, *adjectives* and *compliments* dictionaries, the corresponding binary features are set to true. This implies that the word can be classified as *noun*, *adjective* or as the semantic tag *compliments*.
- Sentence related features: introduced to support the diffusion of relevant pieces of information along the whole sentence. This kind of features is used to identify some potentially relevant semantic elements in the sentence. CRFs natively promote information flow through the graph, however performance improvements were experienced using this kind of features.

5 Experimental Results

As previously explained, only in-domain sentences were used to train the model, but both in-domain and out-of-domain sentences were used to test. For the out-of-domain inputs, the model is expected to choose a correct syntactic parse tree

which includes all the relevant terms in the sentence. The presence of the main concepts in the syntactic tree is a key factor since the ECA will use them to search in the host web page.

To reduce risk of overfitting, a 5-fold cross validation was applied on the in-domain dataset. The in-domain dataset, consisted of 4,096 propositions and was divided into 5 subset of approximately 820 sentences each. Each time, one of the 5 subsets was used to test while the remaining 4 subsets were used to train. In this way, each sentence in the dataset was used to test exactly once and 4 times to train. The k-fold technique for performance estimation is computationally very expensive but allows to obtain a more accurate estimate of true accuracy than classical hold-out methods. Out-of-domain sentences were tested using a model trained with the whole in-domain dataset. The best results have been obtained by setting the CRF's window size to 7; the number of tokens representing context surrounding a word.

An Intel Quad Core Q9400 2.66 GHz machine with 4,096 MB of RAM was used to train the model. Training required a very long time to converge, before introduction of phrase-based features, about 90 hours were necessary to train with the whole in-domain dataset. After the introduction of this kind of features, traning time decreased to about 40 hours. During the experiments, a real risk of local minima was detected.

Performance of both rule based (baseline) and CRFs based reranking systems were evaluated in terms of accuracy, F-measure, precision and recall. The Table 1 shows the baseline rule system performance: rules perform well on in-domain sentences, while on out-of-domain sentences, the performance dramatically drops by losing 13,18% on accuracy and 8,89% on F-measure. Accuracy was calculated among the whole set of 223 categories, while precision, recall and F-measures were calculated only for those categories that occurred more than 20 times in the dataset.

Table 1. Baseline rule-based system performance

	Accuracy	F-measure	Precision	Recall
In-domain	86.59%	92.77%	95.32%	90.35%
Out-of-domain	73.41%	83.88%	85.78%	82.06%
Mixed	80.00%	88.32%	90.55%	86.21%

The CRFs based reranking performance is depicted in Table 2; CRFs perform worse than the baseline system when in-domain sentences are considered, while they perform better than the baseline system when out-of-domain sentences are considered. In this case CRFs significantly improve the baseline system by obtaining a 5.21% increase on accuracy and a 4.98% increase on F-measure.

Table 3 shows the performance evolution in relation to the training set size. Due to considerable computational costs, 5-fold cross validation was applied only for the biggest training set; the remaining were tested with a simple hold-out technique. It is worthwhile that each training set in the table isn't a new training

Table 2. CRFs-based Reranking performance

	Accuracy	F-measure	Precision	Recall
In-domain	80.64%	87.21%	90.49%	84.15%
Out-of-domain	78.62%	88.86%	89.76%	87.98%
Mixed	79.63%	88.03%	90.12%	86.07%

Table 3. Performance evolution on different training set sizes

Training set size	Mixed Accuracy	Mixed F-measure
800	51.74%	59.16%
1,756	64.43%	76.12%
2,480	72.09%	81.70%
3,600	77.09%	87.14%
4,096	79.63%	88.03%

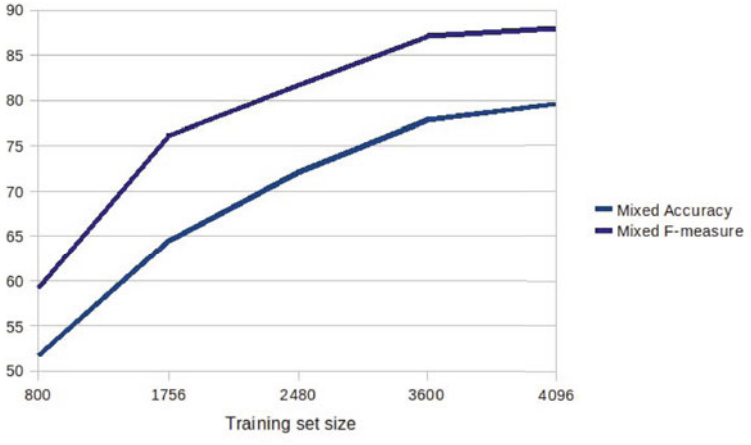


Fig. 5. Performance evolution on different training set sizes

set, but only an extended version of the previous one. Figure 5 shows how the performance improvements are slowly getting smaller as the training set size increase and is essentially stable around 4,000 sentences.

6 Conclusions

The performances achieved by the baseline system and the new proposal are quite mirrored: rule-based performance results are better for the in-domain sentences, while CRFs are better for the out-of-domain ones. The reason why CRFs outperforms the baseline system on out-of-domain sentences is mainly because

they learn which terms are relevant for this particular domain, even if they are to be parsed within a syntactic tree. On the other hand, the baseline system has no semantic knowledge when trying to rerank syntactic parse trees.

The CRFs approach on its own would provide similar results to the baseline system in terms of the overall performance. However, the baseline approach is still more suitable for this particular ECAs application since, as previously explained, in-domain parsing failures are more harmful than out-of-domain ones.

But the work hereby described is not useless. Actually the results presented in the previous section clearly suggest that a combination of both approaches (rule-based for the in-domain sentence and CRFs for the out-of-domain ones) would very much increase the overall performance of the system.

Moreover, the relative importance of both approaches depends on the particular domain evaluated. In section 4.2 a division in-domain versus out-of-domain sentences of 80/20 was detailed. This percentage is very much dependant on the domain and the particular coverage of the ECA application. CRFs based approach would be more suitable for applications with higher out-of-domain input sentences percentage.

7 Future Work

The best way to make use of this approach is by combining it with the baseline rule–based one. There are two alternative approaches to accomplish this:

- Placing a filtering module before both models. This module will decide if the input sentence is an in–domain one, therefore calling the rule based model, or an out–of–domain one, calling the CRFs model.
- Calling the CRFs model always and defining a likelihood threshold *above* which, the CRF solution is discarded and the rule based model is used.

A major concern of the CRFs model described in this paper is the need of a big corpus of input sentences and the man hours needed to tag them. These elements are particularly relevant in the case where the model is to be used for real world applications. Future research directions should focus not only on performance improvement but also on these practical issues.

Acknowledgments. This work has been founded by the Spanish Ministry of Education under grant No TIN2009-14659-C03-03, and the Regional Government of Andalucia under grant No P09-TIC-5138.

References

1. Lester, J., Branting, K., Mott, B.: Conversational Agents. The Practical Handbook of Internet Computing. Chapman and Hall, Boca Raton (2004)
2. Robinson, S., Traum, D., Ittycheriah, M., Henderer, J.: What would you ask a Conversational Agent? Observations of Human-Agent Dialogues in a Museum Setting. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)

3. De Angeli, A., Johnson, G., Coventry, L.: The Unfriendly User: Exploring Social Reactions to Chatterbots. In: Proceedings of International Conference on Affective Human Factor Design, pp. 257–286 (2001)
4. Schulman, D., Bickmore, T.: Persuading users through counseling dialogue with a conversational agent. In: Proceedings of the 4th International Conference on Persuasive Technology (2009)
5. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. ACL (2005)
6. Riezler, S., King, T.H., Kaplan, R.M., Crouch, R., Maxwell, J.T., Johnson, I.M.: Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (2002)
7. Collins, M., Koo, T.: Discriminative Reranking for Natural Language Parsing. In: Computational Linguistics, pp. 175–182. Morgan Kaufmann, San Francisco (2003)
8. Ge, R., Mooney, R.J.: Discriminative Reranking for Semantic Parsing. In: Proceedings of the COLING/ACL-2006 Main Conference Poster Sessions (2006)
9. Quesada J. F., Amores J. G.: Diseño e implementacin de sistemas de traduccin automtica. In: Universidad de Sevilla, Secretariado de publicaciones (2002)
10. Amores, J.G., Quesada, J.F.: Episteme. In: Procesamiento del Lenguaje Natural (1997)
11. Bresnan, J.: The mental representation of grammatical relations. The MIT Press, Cambridge (1982)
12. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data (2001)
13. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE (1989)
14. McCallum, A.K.: Mallet: A machine learning for language toolkit (2002)

Automatic Distractor Generation for Domain Specific Texts

Itziar Aldabe and Montse Maritxalar

IXA NLP Group,
University of the Basque Country, Spain
{itziar.aldabe,montse.maritxalar}@ehu.es
http://ixa.si.ehu.es/Ixa

Abstract. This paper presents a system which uses Natural Language Processing techniques to generate multiple-choice questions. The system implements different methods to find distractors semantically similar to the correct answer. For this task, a corpus-based approach is applied to measure similarities. The target language is Basque and the questions are used for learners' assessment in the science domain. In this article we present the results of an evaluation carried out with learners to measure the quality of the automatically generated distractors.

Keywords: NLP for educational purposes, semantic similarity, distractor.

1 Introduction

The generation of Multiple-Choice Questions (MCQ), one of the measures used for formative assessment, is difficult and time consuming. The implementation of a system capable of generating MCQs automatically would reduce time and effort and would offer the possibility of generating a great amount of questions easily. In our proposal, we use Natural Language Processing (NLP) techniques to construct MCQs integrated in didactic resources.

There are different NLP-based approaches which have proved that the automatic generation of multiple-choice questions is viable. Some of them focus on testing grammar knowledge for different languages, such as English [1] or Basque [2]. Others apply semantic features in order to test general English knowledge [3], [4] or knowledge of specific domains [5]. Our work is focused on the automatic generation of MCQ in a specific domain, i.e. science domain. The target language is Basque.

The objective is to offer experts a helping tool to create didactic resources. Human experts identified the meaningful terms (i.e. words) of a text which were to be the blanks of the MCQs. Then, the system applied semantic similarity measures and used different resources such as corpora and ontologies in the process of generating distractors[1]. The aim of this work is to study different

[1] The incorrect options of the MCQs.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 27–38, 2010.

methods to automatically generate distractors of high quality. That is to say, distractors that correspond to the vocabulary studied by learners as part of the curricula.

As there must be only one possible answer among the options of each MCQ, experts had to discard those distractors that could form a correct answer. Our purpose was to evaluate the system itself by means of an evaluation in a real situation with learners. The results of a test exercise was used to measure the quality of the automatically generated distractors. The evidence provided by the results will be used to improve the methods we propose.

The paper is organised as follows: section 2 explains the scenario to generate and analyse the questions. The methods we have used to generate distractors are explained in section 3. Section 4 presents the experimental settings and section 5 shows the results obtained when evaluating the questions with learners. Finally, section 6 outlines some conclusions and future work.

2 Design of the Scenario

We designed an experiment in which most of the external factors which could have an influence on the evaluation process were controlled.

The multiple-choice questions were presented to learners together with the whole text. Each MCQ is a *stem* and a set of options. The stem is a sentence with a blank. Each blank presents different options, being the correct answer the *key* and the incorrect answers the *distractors*. Example 1 shows an example of MCQs in the context of use.

Example 1. Espazioan itzalkin erraldoi bat ezartzeak, bestalde, Lurrari ...6... egingo lioke, poluitu gabe. Siliziozko milioika disko ...7... bidaltzea da ikertzaileen ideia. Paketetan jaurtiko lirateke, eta, behin diskoak zabalduta, itzalkin-itxurako egitura handi bat osatuko lukete. Hori bai, ...8... handiegiak izango lituzke.[2]

6 a. babes b. aterki c. defentsa d. itzala
7 a. unibertsora b. izarrera c. galaxiara d. espaziora
8 a. kostu b. prezio c. eragozpen d. zailtasun

The process of generating and analysing the questions consists of the following steps:

- Selection of the texts: experts on the generation of didactic resources selected the texts on an specific domain, taking into account the level of the learners and the length of the texts.
- Marking the blanks: the terms to be considered as keys had to be relevant within the text. The marking was carried out manually.
- Generation of distractors: for each stem and key selected in the previous step, distractors were generated.
- Choosing the distractors: experts had to verify that the automatically generated distractors could not fit the blank.

[2] 6 a. protection b. umbrella c. defense d. shadow.

– Evaluation with learners: each learner read the MCQs embedded in a text and chose the correct answer among 4 options.
– Item Analysis: based on learners' responses, an item analysis process was carried out to measure the quality of the distractors.

3 Distractor Generation

When generating distractors, the purpose is to find words which are similar enough to the key but which are incorrect in the context (to avoid the generation of more than one correct answer).

We wanted to generate questions to test the knowledge on a specific domain, i.e. the science domain. The implemented methods are based on *similarity measures*. For that, the system employs the *context* in which the key appears to obtain distractors which are related to the it.

3.1 Word Space: Latent Semantic Analysis

Similarity measures are usual in different NLP applications such us in generating distractors. Two main approaches are used: knowledge-based methods and corpus-based methods. In fact, some researches employ WordNet to measure semantic similarity [4], others use distributional information from the corpus [6] and finally, there are some studies which exploit both approaches [5].

Measuring similarity for minority languages has some limitations. The main difficulty when working with such languages is the lack of resources. In our case, the main knowledge-based resource for Basque [7] is not finished yet: the Basque WordNet[3] is not useful in terms of word coverage, as it has 16,000 less synsets for nouns than WordNet 3.0. As a consequence, we decided to choose as the starting point a corpus-based method to carry out the experiments. Nonetheless, we also used a knowledge-based approach to refine the distractor selection task (cf. Section 3.2).

The system uses context-words to compute the similarity deploying Latent Semantic Analysis (LSA). LSA is a theory and method for extracting and representing the meaning of words [8]. It has shown encouraging results in a number of NLP tasks such as Information Retrieval [9,10] and Word Sense Disambiguation [11]. It has also been applied in educational applications [8] and in the evaluation of synonym test questions [12].

Our system makes use of Infomap software [13]. This software uses a variant of LSA to learn vectors representing the meanings of words in a vector-space known as WordSpace. In our case, it indexes the documents in the corpora it processes and performs word to word semantic similarity computations based on the resulting model. As a result, the system extracts the words that best match a query according to the model.

Build Word Space and Search: As the MCQs we work with are focused on the science domain, the collected corpus consists of a collection of texts related

[3] Nouns are the most developed ones.

to science and technology [14]. The corpus is composed of two parts. For this work, we used the balanced part (3 million words) of the specialised corpus.

In the process of building the model, the matrix was created from the lemmatized corpus. To distinguish between the different categories of the lemmas, the matrix not only took into account the lemma but also its category. The matrix contains nouns, verbs, adjectives and adverbs.

Once we obtained the model based on the specialised corpus, we had to set the context to be searched. After testing different windows, we set the sentence as the context.

3.2 Methods for Distractor Generation

The words found in the model were the starting point to generate the distractors for which different methods can be applied. The baseline method (LSA method) is only based on the output of LSA. The rest of the methods combine the output of the model with additional information to improve the quality of the distractors.

LSA Method: The baseline system provides InfoMap with the whole sentence where the key appears. As candidate distractors the system offers the first words of the output which are not part of the sentence and match the same PoS. In addition, a generation process is applied to supply the distractors with the same inflection form as the key.

LSA & Semantics & Morphology: One of the constraints here is to avoid the possibility of learners' guessing the correct choice by means of semantic and morphology information.

Let us see as an example a question whose stem is *"Istripua izan ondoren, ... sendatu ninduen"* (After the accident, ... cured me) the key is *medikuak* (the doctor) and a candidate distractor is *ospitalak* (the hospital). Both words are related and belong to the same specific domain. Learners could discard *ospitalak* as the answer to the question because they know that the correct option has to be a person in the given sentence. The system tries to avoid this kind of guessing by means of semantic information. Therefore, applying this method, the system does not offer *ospitalak* as a candidate distractor.

The system uses two semantic resources:

a) Semantic features of common nouns obtained with a semiautomatic method [15]. The method uses semantic relationships between words, and it is based on the information extracted from an electronic monolingual dictionary. The extracted semantic features are animate, human, concrete etc. and are linked to the entries of the monolingual dictionary.
b) The Multilingual Central Repository (MCR) which integrates different local WordNets together with different ontologies [16]. Thanks to this integration, the Basque words acquire more semantic information to work with. In this approach, the system takes into account the properties of the Top Concept Ontology, the WordNet Domains and the Suggested Upper Merged Ontology (SUMO).

In a first step, this method obtains the same candidate distractors as the LSA method and then it proposes only those which share at least one semantic characteristic with the key. To do so, the system always tries to find firstly the entries in the monolingual dictionary. If they share any semantic feature, the candidate distractor is proposed; if not, the system searches the characteristics in MCR, which works with synsets. By contrast, the output of Infomap are words. In this approach, we have taken into account all the synsets of the words and checked if they share any characteristic. That is, if a candidate distractor and the key share any characteristic specified by the Top Concept Ontology, the WordNet Domains or SUMO, the candidate distractor is suggested.

One might think that after obtaining distractors which share at least one semantic characteristic with the key, the system does not need any extra information to ensure that they are valid distractors. However, working with all the senses of the words may yield not valid distractors in terms of semantics. Moreover, there are some cases in which two words share a semantic characteristic induced from MCR but which would not be suitable distractors because of their morphosyntax.

In the last step, the method takes the candidate distractors which share at least one semantic characteristic with the key and it takes into account morphosyntax.

Basque is an agglutinative language in which suffixes are added to the end of the words. Moreover, the combination of some morphemes and words is not possible. For instance, while the lemma "ospital" (hospital) and the morpheme "-ko" form the word "ospitaleko" (of the hospital), it is not possible to combine the lemma "mediku" (doctor) with the suffix "-ko", since "-ko" is only used to express the locative genitive case with inanimate words.

As the input text is previously analysed by a morphosyntactic analyser, the system distinguishes the lemma and the morphemes of the key. It identifies the case marker of the key and it generates the corresponding inflected word of each candidate distractor using the lemma of the distractor and the suffix of the key as basis.

Once distractors are generated, the system searches for any occurrence of the new inflected word in a corpus. If there is any occurrence, the generated word becomes a candidate distractor. The searching is carried out in a Basque newspaper corpus which is previously indexed using swish-e[4] to ensure a fast search.

That certain words do not appear in the corpus does not mean that they are incorrect. Those distractors that do appear in the corpus will be given preference over distractors of common usage.

In this step, the system tries to avoid candidate distractors which the learners would reject based on their incorrect morphology.

LSA & Specialised Dictionary: The third method combines the information offered by the model and the entries of an encyclopaedic dictionary of Science and Technology for Basque [17]. The dictionary comprises 23,000 basic concepts related to Science and Technology divided into 50 different topics.

[4] http://swish-e.org/

Based on the candidate distractors generated by the LSA method, the system searches in the dictionary the lemmas of the key and the distractors. If there is an appropriate entry for all of them, the candidate distractors which share the topic with the key in the encyclopaedic dictionary are given preference. Otherwise, the candidate distractors with an entry in the dictionary take preference in the selection process. In addition, those candidates which share any semantic characteristic (cf. 3.2) with the key have preference to be suitable distractors.

LSA & Knowledge-based Method: This method is a combination of corpus-based and knowledge-based approaches to measure the similarities. Similarity is computed in two rounds. First the system selects the candidate distractors based on LSA and then, a knowledge-base structure is used to refine the selection.

The knowledge-based approach [18] uses a graph-based method based on WordNet, where the concepts in the Lexical Knowledge Base (LKB) represent the node in the graph, and each relation between concepts is represented by an undirected edge[5]. Given an input piece of text, this approach ranks the concepts of the LKB according to the relationships among all content words. To do so, Personalized PageRank can be used over the whole LKB graph: given an input text, e.g. a sentence, the method extracts the list of the content nouns which have an entry in the dictionary and relates them to LKB concepts. As a result of the PageRank process every LKB concept receives a score. Therefore, the resulting Personalized PageRank vector can be seen as a measure of the structural relevance of LKB concepts in the presence of the input context. In our case, we use MCR 1.6 as the LKB and Basque WordNet as the dictionary.

The method is defined as follows: Firstly, the system obtains a ranked list of candidate distractors based on the LSA model. Secondly, the Personalized PageRank vector is obtained for the stem and the key. Thirdly, the system applies the graph-based method for 20 candidate distractors in the given stem. Finally, the similarities among vectors computed by the dot product are measured and a reordering of the candidate distractors is obtained.

4 Experimental Settings

A group of experts chose five articles from a web site[6] that provides current and updated information on Science and Technology in Basque. As selection criteria, they focused on the length of the texts as well as the appropriateness to the learners' level. First of all, the experts marked the blanks of the questions and then the distractors were automatically generated. To identify the best method to generate distractors, we designed different experiments where the system applied all the explained methods for each blank and text.

Blanks: Experts who work on the generation of learning material were asked to mark between 15 and 20 suitable terms in five texts to create multiple-choice

[5] The algorithm and needed resources are publicly available at http://ixa2.si.ehu.es/ukb

[6] www.zientzia.net

questions. The blanks were manually marked because the aim of the experiment was to evaluate the quality of the distractors in a real situation. When proceeding, the experts did not follow any particular guidelines but followed the usual procedure[7]. The obtained blanks were suitable in terms of the appropriateness of the science domain and the stems.

In all, 94 blanks were obtained. As we did not give them any extra information for the marking process, experts marked as keys nouns, verbs, adjectives and adverbs. However, our study from a computational point of view aimed at generating nouns and verbs. 69.14% of the obtained blanks were nouns and 15.95% verbs. This shows that the idea of working with nouns and verbs makes sense in a real situation.

Distractors: The distractors were automatically generated for each blank and method. In the case of the nouns, the four mentioned methods were applied and in the case of the verbs, two methods were applied: the LSA method and the LSA & specialised dictionary method[8].

As the distractor generation task is completely automatic, the possibility of generating distractors that are correct in the given context had to be considered. That is why before testing them with learners the distractors were manually checked.

For each method, we provided experts with the first four candidate distractors. We had foreseen to reject the questions which had less than three appropriate distractors. However, in all the cases three valid distractors were obtained. Just 0.95% of the distractors could be as suitable as the key and 3.25% were rejected as dubious.

For each selected text, we obtained four tests (corresponding to the four methods). Moreover, a fifth test was manually made by experts, who created three different distractors for each blank semantically close to the keys. It is important to point out that the experts did not have any information about the distractors obtained from the automatic process. Finally, the manually built tests were compared to the automatically generated ones.

Schools and Learners: Six different schools took part in the experiment. The exercise was presented to the learners as a testing task and the teachers were not familiar with the texts until they handed out the test to their students.

In all, 266 learners of Obligatory Secondary Education (second grade) participated in the evaluation. They had a maximum of 30 minutes to read and complete the test. The test was carried out in paper in order to avoid all noise[9]. 249 of the learners completed the test and their results were used to analyse the items (questions) (see section 5).

After finishing the testing, an external supervisor collected the results of the exercise in situ.

[7] In this step, the evaluation was blind.

[8] We did not apply the remaining methods because the verbs in the Basque WordNet need of manual revision.

[9] We did not want to evaluate the appropriateness of any computer assisted assessment.

5 Results

By means of this evaluation we intended to improve the automatic methods explained in section 3. The item analysis method was the basis of our evaluation.

The item analysis method reviews items qualitatively and statistically to identify problematic items. The difference between both reviews is that the qualitative method is based on experts knowledge and that the statistical analysis is conducted after the items have been given to students. This paper is focused on the statistical analysis. We have used R free software environment[10] for statistical computing and graphics of the learners' results.

5.1 Item Analysis and Distractor Evaluation

The analysis of item responses in a quantitative way provides descriptions of item characteristics and test score properties among others. There are two main theories to address the problem: Classical Test Theory (CTT) and Item Response Theory (IRT). Both statistical theories have been already used in the evaluation of the automatic generation of distractors [3], [5].

In this experiment, we explored item difficulty, item discrimination and distractors' evaluation based on CTT as [5] did. However, the results obtained by them and our results are not comparable since they test the MCQs separately and we test them within a text.

Item difficulty: The difficulty of an item can be described statistically as the proportion of students who can answer the item correctly. The higher the value of difficulty, the easier the item.

Item discrimination: a good item should be able to discriminate students with high scores from those with low scores. That is, an item is effective if those with high scores tend to answer it correctly and those with low scores tend to answer it incorrectly.

The point-biserial correlation is the correlation between the right/wrong scores that students receive on a given item and the total scores that the students receive when summing up their scores across the remaining items. A large point-biserial value indicates that students with high scores on the overall test are also answering the item correctly and that students with low scores on the overall test are answering the item incorrectly. The point-biserial correlation is a computationally simplified Pearson's r between the dichotomously scored item and the total score. In this approach, we use the corrected point-biserial correlation. That is, the item score is excluded from the total score before computing the correlation. This is important because the inclusion of the item score in the total score can artificially inflate the point-biserial value (due to correlation of the item score with itself).

There is an interaction between item discrimination and item difficulty. It is necessary to be aware of two principles: very easy or very difficult test items have little discrimination and items of moderate difficulty (60% to 80% answering

[10] http://www.r-project.org/

correctly) generally are more discriminating. Item difficulty and item discrimination measures are useful only to help to identify problematic items. Poor item statistics of the results should be put down to ineffective distractors.

Distractor evaluation: to detect poor distractors, the option-by-option responses of high-scored and low-scored learners groups were examined. To this purpose, two kind of results will be presented in the next section: the number of distractors never chosen by the learners and a graphical explanation of the used distractors.

5.2 Interpreting the Results of the Tests

Table 1 shows the average of item difficulty and item discrimination results obtained for all the items in a text. The table shows the results for the manually and automatically generated tests.

In the case of item difficulty, each row presents the item difficulty index together with the standard deviation, as well as the percentage of easy and difficult items. In this work, we have marked an item to be easy if more than 90% of the students answer it correctly. On the other hand, an item is defined as difficult when less than 30% of the students choose the correct answer. The results shown for the manually generated test are promising (near 0.5), and there is not significant differences among the automatic methods. All of them tend to obtain better results with the second text and tend to create easy items.

The results of item discrimination take into account the responses of the high-scoring and low-scoring students. The high-scoring group is the top 1/3 of the class, and the low-scoring group comprises students with test scores in the bottom 1/3 of the class. Regarding item discrimination, the corrected point-biserial index with its standard deviation as well as the percentage of items with negative values are shown in the table.

Even though all the results obtain a positive mean (a value of at least 0.2 is desirable), in 8 out of the 10 cases negative point-biserial indexes are obtained. These negative values represent the percentage of items correctly responded by a higher number of low-scored students than high-scored ones. To identify the reasons underlying these results we study the option-by-option responses of high-scored and low-scored groups. Such study led us to evaluate the distractors themselves.

Figure 1 shows in a graphic way the distribution of the distractors among the low-scored and high-scored groups. The x axe represents the number of low-scored students selecting a distractor and the y axe represents the number of high-scored ones. In this experiment we have studied the results related to 108 distractors, limiting the number of students per test to 20.

Regarding the number of different distractors, in the case of the manually generated distractors, 83 (76.85%) out of the 108 distractors were chosen. In the cases of the automatically generated distractors the results were 64 (59.26%) for the LSA method, 54 (50.00%) for the LSA & semantics & morphology method, 67 (62.04%) for the LSA and & encyclopaedic dictionary method and 60 (55.56%) for the LSA & knowledge-based method.

Table 1. Item Analysis

		Difficulty			Item Discrimination	
		Item Difficulty	Easy	Difficult	Corrected Point-biserial	Neg.
LSA	Text1	0.79 (±0.18)	29.41%	0.00%	0.22 (±0.25)	17.65%
	Text2	0.67 (±0.20)	5.26%	5.26%	0.10 (±0.21)	31.58%
LSA & sem. & morph.	Text1	0.83 (±0.15)	35.29%	0.00%	0.26 (±0.16)	0.00%
	Text2	0.71 (±0.21)	21.05%	10.53%	0.30 (±0.15)	5.26%
LSA & spec. dictionary	Text1	0.70 (±0.23)	17.65%	11.76%	0.22 (±0.14)	5.88%
	Text2	0.66 (±0.22)	5.26%	5.26%	0.22 (±0.35)	26.32%
LSA & Knowledge-based	Text1	0.76 (±0.22)	23.53%	11.76%	0.33 (±0.30)	11.76%
	Text2	0.68 (±0.19)	26.32%	15.79%	0.41 (±0.19)	0.00%
Manually generated	Text1	0.66 (±0.23)	0.00%	5.88%	0.13 (±0.21)	23.53%
	Text2	0.46 (±0.26)	0.00%	36.84%	0.14 (±0.21)	21.05%

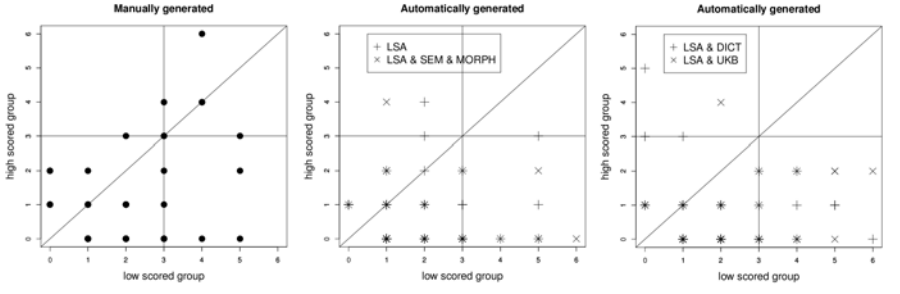


Fig. 1. Distractors Evaluation. * is used when both methods share the point.

Based only on the selected distractors, this last method gives the best results in relation to the percentage of distractors that discriminates positively among the low and high-scored groups: 90.00% (54 out of 60). The distractors obtained by the LSA & semantics & morphology method discriminated positively in 87.04% of the cases, the LSA & dictionary method in 79.10% of the cases, the LSA method in 76.56% of the cases and the manual method in 75.90% of the cases. In a graphic way, the distribution of the low-right side of the graphics can be interpreted as the set of good distractors.

The distribution of the high-left side of the graphics represents distractors that have to be revised because they confuse high-scored students and do not confuse low-scored learners. The reason could be that low-scored learners have not enough knowledge to be confused. Looking at the results of the methods, the LSA method tend to confuse more than the other methods (14.06%), followed by the manual method (12.05%), the LSA & dictionary method (11.94%), the LSA & semantics & morphology method (7.41%) and the LSA & knowledge-based method (6.67%).

It seems there is a relation between the number of the selected distractors and the percentage of discrimination: the lower the number of distractors, the higher the positive discrimination. However, the LSA method does not follow this assumption.

In order to improve the methods, we are planning to study in more depth the distractors that were never chosen. Moreover, it is also necessary to analyse on two other aspects: the domain nature and the part-of-speech of the keys. We must not forget that experts marked the blanks without being instructed. Therefore the blanks did not have to correspond with words related to the specific domain.

6 Conclusions and Future Work

The article presents a study about automatic distractor generation for domain specific texts. The system implements different methods to find distractors semantically similar to the key. It uses context-words to compute the similarity deploying LSA. We have used a balanced part of a specialised corpus to build the model. In the near future we will make use of the whole specialised corpus to model it.

In this approach, we have explored item difficulty, item discrimination and distractors' evaluation based on Classical Test Theory. The results shown for the manually generated test were promising, and there were not significant differences among the methods. The item discrimination measure led us to study the option-by-option responses of high-scored and low-scored groups and we finished the study with the evaluation of the distractors. Such evaluation gave us evidence to improve the methods regarding the domain nature and part-of-speech of the keys, and the need to enlarge the context when applying LSA. In addition, we are planning to test the distractors with more learners. Finally, the fact that the distractors tend to confuse high-scored learners, but not low-scored learners needs of deeper analysis.

In our opinion, working in a specific domain may improve the quality of the distractors so in the near future we will design new experiments with test exercises independent from the domain to compare the results with the ones obtained in the current study.

For future work we are also planning to use data mining techniques to identify the blanks of the text. Finally, reliability measures should also be considered in future research. Reliability tells us whether a test is likely to yield the same results if administered to the same group of test-takers multiple times. Another indication of reliability is that the test items should behave the same way with different populations of test-takers.

Acknowledgments. We would like to thank to the institution named "Ikastolen Elkartea" who has assigned good experts in order to work on the tasks of this work. This research is being supported by the Basque Government (SAIOTEK project, S-PE09UN36 and ETORTEK project, IE09-262).

References

1. Hoshino, A., Nakagawa, H.: Assisting cloze test making with a web application. In: Proceedings of SITE (Society for Information Technology and Teacher Eduation), San Antonio, U.S., pp. 2807–2814 (2007)

2. Aldabe, I., Lopez de Lacalle, M., Maritxalar, M., Martinez, E., Uria, L.: ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 584–594. Springer, Heidelberg (2006)
3. Sumita, E., Sugaya, F., Yamamota, S.: Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions. In: 2nd Workshop on Building Educational Applications Using NLP (2005)
4. Pino, J., Heilman, M., Eskenazi, M.: A Selection Strategy to Improve Cloze Question Quality. In: Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains (2008)
5. Mitkov, R., Ha, L.A., Varga, A., Rello, L.: Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In: Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics, pp. 49–56 (2009)
6. Smith, S., Kilgarriff, A., Sommers, S., Wen-liang, G., Guang-zhong, W.: Automatic Cloze Generation for English Proficiency Testing. In: Proceeding of LTTC conference, Taipei (2009)
7. Agirre, E., Ansa, O., Arregi, X., Arriola, J.M., Diaz de Ilarraza, A., Pociello, E., Uria, L.: Methodological issues in the building of the Basque WordNet: quantitative and qualitative analysis. In: Proceedings of the first International WordNet Conference, Mysore, India (2002)
8. Landauer, T.K., McNamara, D.S., Dennis, S., Kintsch, W.: Handbook of Latent Semantic Analysis. Lawrence Erlbaum Associates, Mahwah (2007)
9. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
10. Gliozzo, A.M., Giuliano, C., Strapparava, C.: Domain Kernels for Word Sense Disambiguation. In: 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005). University of Michigan, Ann Arbor (2005)
11. Schütze, H.: Automatic word sense discrimination. In: Computational Linguistics, vol. 24(1), pp. 97–124 (1998)
12. Turney, P.: Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 491–502. Springer, Heidelberg (2001)
13. Dorow, B., Widdows, D.: Discovering corpus-specific word senses. In: Proceeding of EACL, Budapest (2003)
14. Areta, N., Gurrutxaga, A., Leturia, I., Alegria, I., Artola, X., Diaz de Ilarraza, A., Ezeiza, N., Sologaistoa, A.: ZT Corpus: Annotation and tools for Basque corpora. In: Copus Linguistics, Birmingham, UK (2007)
15. Diaz de Ilarraza, A., Mayor, A., Sarasola, K.: Semiautomatic labelling of semantic features. In: 19th International Conference on Computational Linguistics (2002)
16. Atserias, J., Villarejo, L., Rigau, G., Agirre, E., Carroll, J., Magnini, B., Vossen, P.: The MEANING Multilingual Central Repository. In: Proceedings of the Second International WordNet Conference-GWC, Brno, Czech Republic, pp. 23–30 (2004)
17. Zerbitzuak, E.H. (ed.): Elhuyar Zientzia eta Teknologiaren Hiztegi Entziklopedikoa. Elhuyar Edizioak/Euskal Herriko Unibertsitatea (2009)
18. Agirre, E., Soroa, A.: Personalizing PageRank for Word Sense Disambiguation. In: Proceedings of EACL 2009, Athens, Greece, pp. 33–41 (2009)

Summarization as Feature Selection for Document Categorization on Small Datasets

Emmanuel Anguiano-Hernández[1], Luis Villaseñor-Pineda[1], Manuel Montes-y-Gómez[1], and Paolo Rosso[2]

[1] Laboratory of Language Technologies, Department of Computational Sciences, National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico
{eanguiano,villasen,mmontesg}@inaoep.mx
[2] Natural Language Engineering Lab, ELiRF, Department of Information Systems and Computation, Polytechnic University of Valencia (UPV), Spain
prosso@dsic.upv.es

Abstract. Most common feature selection techniques for document categorization are supervised and require lots of training data in order to accurately capture the descriptive and discriminative information from the defined categories. Considering that training sets are extremely small in many classification tasks, in this paper we explore the use of unsupervised extractive summarization as a feature selection technique for document categorization. Our experiments using training sets of different sizes indicate that text summarization is a competitive approach for feature selection, and show its appropriateness for situations having small training sets, where it could clearly outperform the traditional information gain technique.

Keywords: Text Categorization, Text Summarization, Feature Selection.

1 Introduction

Automatic document categorization is the task of assigning free text documents into a set of predefined categories or topics. Currently, most effective solutions for this task are based on the paradigm of supervised learning, where a classification model is learned from a given set of labeled examples called training set [7]. Within this paradigm, an important process is the identification of the set of features (words in the case of text categorization) more useful for the classification. This process, known as feature selection, tends to use statistical information from the training set in order to identify the features that better describe the objects of different categories and help discriminating among them. Due to the use of that statistical information, the larger the training set, the better the feature selection. Unfortunately, due to the high costs associated with data labeling, for many applications these datasets are very small. Because of this situation it is of great importance to search for alternative feature selection methods specially suited to deal with small training sets.

In order to tackle the above problem, in this paper we propose to apply unsupervised extractive summarization as a feature selection technique; in other words, we propose reducing the set of features by representing documents by means of a representative subset of their sentences. Our proposal is supported on two facts about

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 39–44, 2010.

extractive summarization. First, it has demonstrated to capture the essence of texts by selecting their most important sections, and, consequently, a subset of words adequate for their description. Second, it is an inherently local process, where each document is reduced individually, bypassing the restrictions imposed by the size of the given training set.

Through experiments on a collection consisting of three training sets of different sizes we show that text summarization is a competitive approach for feature selection and, what is more relevant, that it is specially appropriate for situations having small training sets. Particularly, in this situations the proposed approach could significantly improved the results achieved by the information gain technique.

The rest of this document is organized as follows. Section 2 presents some related works concerning the use of text summarization in the task of document categorization. Section 3 describes the experimental platform; particularly it details the feature selection process and the used datasets. Then, Section 4 shows the results achieved by the proposed approach as well as some baseline results corresponding to the application of information gain as feature selection technique. Finally, Section 5 presents our conclusions and future work ideas.

2 Related Work

Some previous works have considered the application of text summarization in the task of document categorization. Even though these works have studied different aspects of this application, most of them have revealed, directly or indirectly, the potential of text summarization as a feature selection technique.

Some of these works have used text summarization (or its underlying ideas) to *improve the weighting of terms* and, thereby, the classification performance. For instance, Ker and Chen [2] weighted the terms by taking into account their frequency and position in the documents; whereas Ko et al. [3] considered a weighting scheme that rewards the terms from the phrases selected by a summarization method.

A more ambitious approach consists of applying text summarization with the aim of reducing the representation of documents and *enhancing the construction of the classification model*. Examples from this approach are the works by Mihalcea and Hassan [5] and Shen et al. [8]. The former work is of special relevance since it showed that significant improvements can be achieved by classifying extractive summaries rather than entire documents.

Finally, the work by Kolcz et al. [4] explicitly proposes the use of summarization *as a feature selection technique*. They applied different summarization methods –based on the selection of sentences with the most important concentration of keywords or title words– and compared the achieved results against those from a statistical feature selection technique, concluding that both approaches are comparable.

Different to these previous works, this paper aims to determine the usefulness of summarization as feature selection technique for the cases consisting of small training sets. Our assumption is that, because summarization is a local process, done document by document without considering information from the entire dataset, it may be particularly appropriate for these cases. Somehow, our intention is to extent the conclusions by Kolcz et al. by showing that, although summarization and statistical feature

selection techniques are comparable for most of the cases, the former is a better option for situations restricted by the non-availability of large training sets.

3 Experimental Platform

3.1 Feature Selection Process

Because of our interest to evaluate the effectiveness of text summarization as a feature selection technique, we compared its performance against the one of a traditional supervised (statistical) approach. Particularly, to summarize the documents we used the well-known $HITS_A$ *directed backward* graph-based sentence extraction algorithm [6]. The choice of this algorithm was motivated by its relevant results in text summarization as well as by its previous usage in the context of document categorization [5]. On the other hand, we considered the *information gain* (IG) measure as exemplar from supervised techniques [9].

In a few words, the feature selection was carried out as follows:

1. Summarize each document from the training set, by selecting the $k\%$ of their most relevant sentences, in line with the selected summarization method.
2. Define the features as the set of words extracted from the summaries, eliminating the stop words.

In contrast to this approach, the common (statistical) feature selection process defines the features as the set of words having positive information gain ($IG > 0$) within the entire dataset. That is, it selects the words whose presence or absence gives the larger information for category prediction.

3.2 Evaluation Datasets

For the experiments we used the *R8 collection* [1]. This collection is formed by the eight largest categories from the Reuters-21578 corpus, which documents belong to only one class. It contains 5189 training documents and 2075 test documents.

With the aim of demonstrating the appropriateness of the proposed approach for situations having small training sets, we constructed two smaller collections from the original R8 corpus: R8-41 and R8-10, consisting of 41 and 10 training documents per class respectively. These collections contain 328 and 80 training documents and the original 2075 test documents. Details can be found in Table 1.

Table 1. Documents distribution in different data sets

Class	R8 Training Set	R8-41 Training Set	R8-10 Training Set	Test Set
earn	2701	41	10	1040
acq	1515	41	10	661
trade	241	41	10	72
crude	231	41	10	112
money-fx	191	41	10	76
interest	171	41	10	73
ship	98	41	10	32
grain	41	41	10	9

4 Results

In the experiments we evaluated the effectiveness of feature selection by means of the classification performance. Our assumption is that, given a fixed test set and classification algorithm, the better the feature selection the higher the classification performance. In particular, in all experiments we used a Support Vector Machine as classification algorithm, term frequency as weighting scheme, and the classification accuracy and micro-averaged F_1 as evaluation measures.

Table 2 shows two baseline results. The first one corresponds to the usage of all words as features (i.e., without applying any feature selection method, except by the elimination of stop words); whereas, the second concerns the usage of only those words having positive information gain. From these results it is clear that the IG-based approach is pertinent for situations having enough training data, where it could improve the accuracy in 1.5%. However, it is also evident that it has severe limitations to deal with small training datasets. For instance, it only could define 20 relevant features for the R8-10 collection (which represented just 1% of the whole set of words), causing a decrement in the classification accuracy of around 50%.

Table 2. Baseline results: without feature selection and using the information gain criterion

	R8			R8-41			R8-10		
	Features	***Accuracy***	***F1***	***Features***	***Accuracy***	***F1***	***Features***	***Accuracy***	***F1***
All features	17,336	85.25	.842	5,404	78.75	.782	2,305	71.71	.702
IG > 0	1,691	86.51	.857	54	42.89	.539	20	35.57	.0402

Table 3 and table 4 show results from the proposed method for different summary sizes, ranging from 10% to 90% of the original sentences of the training documents. The achieved results are encouraging; they show that text summarization is a competitive approach for feature selection and, what is more relevant, that it is especially appropriate for situations having small training sets. In particular, for the reduced collections R8-41 and R8-10, very small summaries (from 10% to 50%) could outperform, with statistical significance, the results obtained by the application of the IG-based approach ($IG > 0$) as well as those obtained using all words as features. We evaluated statistical significance of the results using the z-test with a confidence of the 95%.

Table 3. Results accuracy of proposed method using summaries of different sizes

	R8			R8-41			R8-10		
Sum. Size	***Number features***	***Our method***	***Top IG***	***Number features***	***Our method***	***Top IG***	***Number features***	***Our method***	***Top IG***
10%	8,289	87.13	85.45	1,943	83.47	80.43	706	76.77	52.24
20%	9,701	88.53	85.54	2,445	82.27	78.02	902	70.17	56.87
30%	11,268	89.20	85.78	3,089	82.89	78.31	1,178	64.67	52.34
40%	12,486	87.90	85.78	3,569	83.52	78.60	1,392	75.23	54.07
50%	13,326	87.42	85.88	3,919	81.40	79.13	1,523	75.08	64.10
60%	14,560	86.89	85.64	4,348	79.66	78.89	1,722	69.40	67.52
70%	15,626	86.75	85.54	4,671	80.10	78.94	1,890	69.73	69.69
80%	16,339	86.70	85.69	5,004	80.43	78.31	2,082	71.23	69.83
90%	17,063	86.27	85.35	5,263	78.89	78.60	2,230	72.58	71.66

In order to have a deep understanding of the capacity of the proposed method, we compared its results against those from a classifier trained with the same number of features but corresponding to the top IG values (indicated in Table 4 as Top-IG). As can be noticed our method always obtain better results, indicating that the information gain cannot be properly evaluated from small training sets. Regarding this fact, it is interesting to notice that for the R8-10 collection, our method allowed a 7% of accuracy improvement (from 71.71 to 76.77) by means of a 70% feature reduction (from 2,305 to 706), whereas, for the same compression ratio, the features selected by their *IG* value caused a 28% drop in the accuracy (from 71.71 to 52.24).

Table 4. F1-measure of the proposed method using summaries of different sizes

	R8		**R8-41**		**R8-10**	
Sum. Size	***Our method***	***Top IG***	***Our method***	***Top IG***	***Our method***	***Top IG***
10%	.876	.846	.842	.817	.776	.572
20%	.886	.846	.834	.790	.709	.659
30%	.891	.848	.836	.789	.654	.618
40%	.877	.848	.842	.789	.766	.631
50%	.870	.848	.819	.791	.763	.700
60%	.864	.846	.798	.787	.683	.717
70%	.862	.845	.800	.786	.685	.716
80%	.861	.847	.803	.780	.693	.698
90%	.856	.843	.784	.781	.712	.703

5 Conclusions and Future Work

This paper studied the application of automatic text summarization as a feature selection technique in the task of document categorization. Experimental results in a collection having three training sets of different sizes indicated that summarization and information gain (a statistical feature selection approach) are comparable when there are enough training data (such as in the original R8 collection), whereas the former is a better option for situations restricted by the non-availability of large training sets (as in the cases of R8-41 and R8-10 collections). This behavior is because summarization is a local process, where each document is reduced individually without considering the rest of the documents; while statistical techniques such as IG require lots of training data in order to accurately capture the discriminative information from the defined categories. Due to this characteristic, as future work we plan to examine the pertinence of summarization-based feature selection into a semi-supervised text classification approach.

It is important to mention that the success of summarization depends on the nature of the documents. In this paper we evaluated the proposed method in a collection of news reports demonstrating its usefulness. As future work we plan to determine its appropriateness for other kinds of documents such as web pages and emails.

Acknowledgments. This work was done under partial support of CONACYT (Project grants 83459, 82050, 106013, and scholarship 271106), and the MICINN TIN2009-13391-C04-03 (Plan I+D+i) TEXT-ENTERPRISE 2.0 research project.

References

1. Cardoso-Cachopo, A.: Improving Methods for Single-Label Text Categorization. PhD Thesis. Technical University of Lisboa, Portugal (October 2007)
2. Ker, S.J., Chen, J.N.: A Text Categorization Based on a Summarization Technique. In: ACL 2000 Workshop on Recent Advances in Natural Language Processing, Honk Kong (2000)
3. Ko, Y., Park, J., Seo, J.: Improving Text Categorization Using the Importance of Sentences. Information Processing and Management (40) (2004)
4. Kolcz, A., Prabakarmurthi, V., Jugal, K.: Summarization as a Feature Selection for Text Categorization. In: Tenth International Conference on Information and Knowledge Management (CIKM 2001), Atlanta GA, USA (2001)
5. Mihalcea, R., Hassan, S.: Using the Essence of Texts to Improve Document Classification. In: RANLP 2005, Borovetz, Bulgaria (2005)
6. Mihalcea, R.: Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. In: ACL 2004, Barcelona, Spain (2004)
7. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Survey 34(1), 1–47 (2002)
8. Shen, D., Yang, Q., Chen, Z.: Noise Reduction through Summarization for Web-Page Classification. Information Processing and Management (43) (2007)
9. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of ICML 1997, pp. 412–420 (1997)

A Formal Ontology for a Computational Approach of Time and Aspect

Aurélien Arena and Jean-Pierre Desclés

LaLIC (Langues, Logiques, Informatique et Cognition)
University of La Sorbonne
Maison de la recherche, 28 rue Serpente, Paris 75006, France

Abstract. This paper provides a linguistic semantic analysis of time and aspect in natural languages. On the basis of topological concepts, notions are introduced like the basic aspectual opposition between event, state and process or that of time of utterance (for the treatment of deictic categories) that are used to analyse the semantics of grammatical tenses or more general situations. This linguistic model yields a conceptualisation reused for the definition of a formal ontology of time and aspect. This ontology is provided with a formal framework based on type theory that enables the processing of temporal and aspectual semantic entities and relations.

Keywords: aspect, time, semantics, formal ontology, type theory, knowledge representation.

1 Introduction

This paper addresses the problem of minimizing the distance between 1) time and aspect *conceptualized from natural languages*, and 2) a computational model enabling a formal treatment of the semantics of texts. Section 1 gives general information about time and aspect and introduces a specific linguistic model resting on topological concepts and taking into account notions like temporal deixis or primitive aspectual values such as state, process or event and derived ones like resultative state (see [1]). Section 2 provides an original formal framework using an applicative language with Church functional types to express this linguistic ontology and then defines concepts and relations of this formal ontology of time and aspect. This work intends to reach a greater expressivity compared to time notions investigated for instance within modal logic like with tense logic, LTL or ITL (if natural language semantics description is the goal to reach), because it develops an interval-based semantics integreting aspectual properties more suitable for natural languages analysis. On the other hand, the reasoning aspect is not considered here. This paper is a knowledge representation investigation leading to a finer expressivity for applications like ontologies population or formal definition of grammatical operators.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 45–56, 2010.

2 An Analysis of Aspect and Temporality

Time notions conceptualized from natural languages are often classified by linguistics into two main concepts, that of time and that of aspect. The former deals with locating situations in time with respect to the time of utterance (e.g. deictic references like *yesterday*) or other situations in time not related to it (e.g. *after the war*). The latter can be defined, following [2], as the parameter that studies the *different ways of viewing the internal temporal constituency of a situation*. Those definitions as it is going to be explained in the next sections can be given a more precise characterization adopting a theoretical point of view. For the present purpose (building an ontology of time and aspect), first a list of fundamental entities that can be found in the semantics of aspect and time is drawn up and then relations over them are identified (according to a specific theory that is developed here).

2.1 Linguistic Concepts for a Model of Aspect and Time

Discussing the different linguistic theories of aspect is not the main purpose of this paper, simply a brief overview of a few typologies of aspectual semantic values will be provided, then some comments are given on how aspectual values in languages can be conveyed by linguistic elements and finally, the theoretical linguistic concepts adopted here are laid out.

Before introducing theoretical elements some linguistic examples of aspectual[1] oppositions are considered without committing to any particular classification.

1. (a) John walked
   (b) John was walking
   (c) John has walked
   (d) John walked to the station

Dealing with aspect, the classification of verbs made by the philosopher Vendler has often been taken up, discussed and also refined. He provided a set of four classes of verbs based on their distinct aspectual properties along with basic linguistic tests to determine if a verb belongs to a given class. The four classes are the following: activity (e.g. *run, walk*), achievement (e.g. *see, reach the top*), accomplishment (e.g. *run a mile, build a house*) and state (e.g. *believe, to be in the room*). For more details about this classification, see [3]. Later, this aspectual typology has been refined on several points, notably the semantic definitions of the aspectual classes. Indeed, according to different semantic features, other classifications are proposed, for instance in ([4], [5], [6]). Mourelatos introduces a more fundamental aspectual distinction between event, state and process from which Vendler's classes can be expressed in a hierarchical way.

Another major refinement about the Vendler classification concerns the elements it applies to. As it has been argued by Dowty, Verkuyl or Mourelatos, an aspectual semantic classification should not be restricted to verbs but rather

[1] Regardless of temporal information for the moment.

to situations described by whole sentences, namely verbs along with their arguments and other aspectual modifiers like adverbials[2] . Consider the following examples.

2. (a) Paul drinks beer.
   (b) Paul drinks a beer.
   (c) Paul read a book in one hour.
   (d) Paul read a book for one hour.

The linguistic theory of time and aspect that is worked out here is now introduced with its formal framework. The overall theory is described in ([9], [10]). Definitions of aspectual semantic values rest on topological concepts, and more precisely on topological intervals of instants (open, closed, half-open) for which topological properties have a linguistic meaning. This interval-based semantics is in line with a model-theoretic approach. In other words, linguistic expressions (to which a logical representation will be given) are given a truth value with respect to topological intervals[3] of the model. We give now information about entities (or linguistic concepts) being taken into account in the model.

Dealing with time in natural languages, the notion of temporal reference system (see [1]) is useful to understand its semantics. Different types of temporal reference systems are defined, first, that created by the utterance process and from which other situations can be located: the enunciative reference system, it is said to be actualized (e.g. *now, yesterday*), secondly, that being not actualized (e.g. *the first of March*), or that of possible situations (e.g. *if I were rich ...*). In the model, by hypothesis the linguistic time is defined as an organized set of temporal reference systems, and each temporal reference system being a continuous and ordered set of instants. They are structured by three relations, the relation of identification ($=$), that of differentiation ($\neq$) for "before" and "after" relations and the breaking relation ($\sharp$) meaning that a temporal reference system is not located with respect to the enunciative reference system like with for instance the marker *once upon a time*.

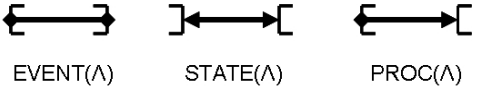
Regarding aspect, the model of this theory is based on three primitive semantic values, those of *process, event* and *state*, and to each is given a conceptual content (analyzed from linguistic data and cognitive considerations). A situation has the value of a process when it expresses an ongoing action without last

[2] Linguistic markers being variable from one language to another. For a theoretical introduction on aspect with examples taken from different languages, see [2].

Dependencies between aspectual values of a sentence and its constituents is known as the aspectual composition phenomena. For instance, see [7] for a treatment of aspect in relation with the different semantic types of nominal arguments (as in example 2.a and 2.b), or [8] which notably takes into account adverbials (as in 2.c and 2.d).

[3] Many linguists have argued about the necessity of intervals for the semantics of time (Bennet, Culioli, Partee, Desclés, Dowty, Kamp), but also in logics (Van Benthem, Montanari [11], Goranko) or in other fields like in IA or philosophy. Generally in opposition to an instant-based ontology.

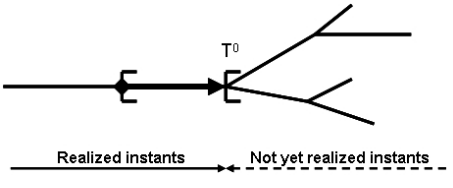
instant. It is formally represented by a right-open and left-closed interval (there is an initial but no final instant). A situation is a state when it expresses an absence of modifications and its semantics is represented by an open interval. On the other hand, a situation is an event when it refers to a modification between two static situations and it is represented by a closed interval (with an initial and a final instant). For instance, a situation having the aspectual value of state is said to be realized onto an open interval (true for every instant belonging to the given interval in the model). Situations having aspectual values are given an underlying logical expression, namely an applicative expression ASP(Λ) where ASP is a class of aspectual operators {PROC, STATE, EVENT} applying to Λ being a timeless[4] predicative relation, those operator yielding aspectualised predicative relation realized on topological intervals.



Up to now entities (state, event, process, temporal referential) of the model have been defined, now relations that can be predicated over them are going to be investigated.

2.2 Linguistic Relations for a Model of Aspect and Time

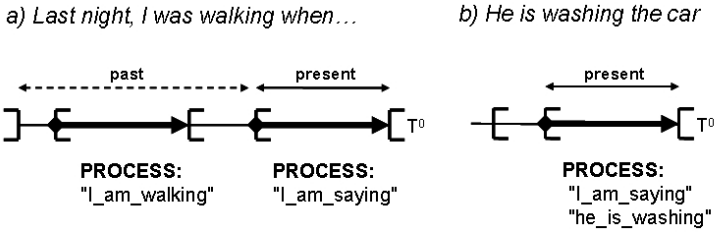
Before introducing temporal relations, a preliminary remark is made on the notion of utterance time that is defined by the utterance process and realized on a right-open interval (not a single instant but a process because uttering takes time and is unaccomplished), see again [1]. The right boundary of this process is written T^0 and is not the last instant of the utterance process but the first instant of the non-realized instants. In terms of the temporal structure, the bound T^0 of the utterance process separates the past from the future in an asymmetric way, the future having a non linear structure. This is one of the reasons why linguistic time can't be conceptualized as a linear order.



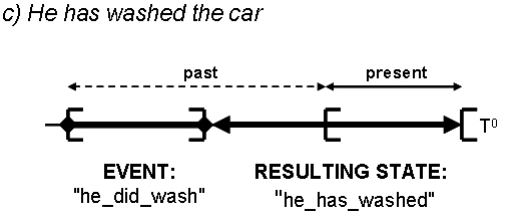
Thus, temporal relations are expressed by precedence or identification[5] relation holding between a situation (with an aspectual value) and the utterance process. Consider some linguistic examples and their corresponding intervals.

[4] Without temporal or aspectual value. More precisely, Λ refers to the notion of *lexis* or *dictum*, denoting "what is said".

[5] Relations between topological intervals are defined formally in the next section.



Aspectual values such as defined above are said to be primitive to the extent that they can be combined to express derived aspectual semantic values like the perfect. A sentence like:



expresses a situation where an event and a state are in a specific configuration. The bound between the event and the state is defined as a continuous cut (as defined by Dedekind[6]), and the adjacent (to the event) resulting state refers to a causal relation holding between both intervals, which correspond to the semantics of the perfect.

As defined in the model, aspectual semantic values are not indeed independent from each other. Consider some simple examples like:

3. (a) He was washing the car.
   (b) He washed the car.
   (c) He has washed the car.

Those situations all refer to a common telic predicative relation (that has to reach a final term to be true, here the right bound of the event). The clause a. refers to an underlying unaccomplished process (right bound open). Once this process is achieved (right bound reached), it is turned into an event, the clause b. From this event can be related some causal situations expressed by a perfect value as in the clause c. Those aspectual properties lead to a general network of dependencies introduced in ([1]).

[6] A continuous cut written t_c. For a set of instants E linearly oriented and such that for A1 and A2 two subsets of E, the following conditions are verified (1) $A1 \cup A2 \supseteq E$, (2) $A1 \cap A2 = \varnothing$ and (3) $A1 < A2$. t_c is a continuous cut of E when either ($t_c \in A1$ and $t_c \notin A2$) or ($t_c \notin A1$ and $t_c \in A2$), exclusively.

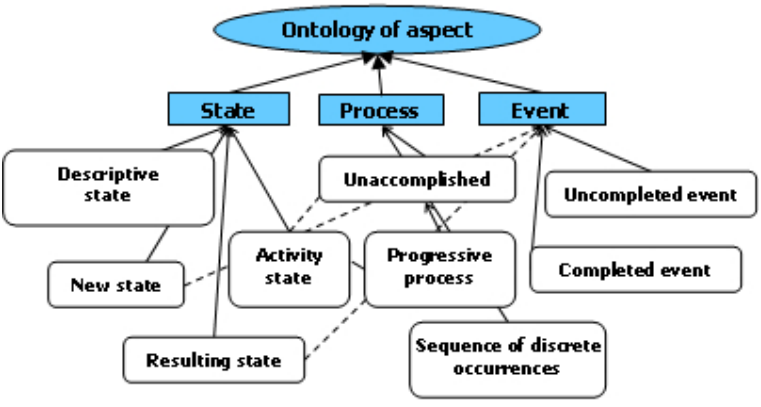


Fig. 1. The core information of the ontology consists of a set of constraints between aspectual values. Arrows in this network can either mean "is a sort of" like in the proposition an activity state is a sort of state, or "implies" like in an event implies a resulting state or "contains" like in a progressive process is contained in an activity state (e.g. the plane is flying vs. the plane is in fly). See [10] for further details and a linguistic account for those relations.

These theoretical linguistic developments are in line with an ontological investigation to the extent that it answers to some questions about the time nature (e.g. analysis of intervals bounds, constraints and relation between them). Now are introduced some tools to formally represent this linguistic ontology.

3 A Formal Ontology of Time and Aspect

First a specific framework is laid out, to make possible the expression of aspectual semantics introduced in the previous section.

3.1 Formal Framework

According to [12], there exists in the literature of knowledge representation several meanings for the term ontology. One meaning identifies an ontology to a specific conceptualization of a knowledge base (here, the formal semantics of time and aspect introduced in section 1). Another meaning concerns the formal account[7] given to such conceptualization (which is the topic of this section).

The formalism that is used here is that of applicative systems with types. Applicative formalisms have been developed along with combinatory logics by Curry who introduced the notion of operator and the basic operation of application. The notion of type here is that of functional types introduced by Church.

[7] For instance description logics, first order logic or applicative systems like combinatory logic or lambda-calculus.

Thus, the basic structure for such system rests on the fundamental distinction between operator and operand. The former being applied to the later to return a result. Each of those three entities having a specific functional type. This notion of type is used to characterize classes of objects operators can be applied to. The construction rule for functional types is the following:

$$\begin{array}{l}(1)\ \text{Primitive types are Types}\\ (2)\ \text{If } \boldsymbol{\alpha} \text{ and } \boldsymbol{\beta} \text{ are Types, then F}\boldsymbol{\alpha\beta} \text{ is a Type .}\end{array} \tag{1}$$

F is the functional type constructor and F$\boldsymbol{\alpha\beta}$ is the type of an operator that can be applied only to operands having the type $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is the type of the result. The application rule is the following:

$$\frac{X : F\boldsymbol{\alpha\beta} \qquad Y : \boldsymbol{\alpha}}{XY : \boldsymbol{\beta}} \ . \tag{2}$$

An ontology being often defined as a set of concepts with relations over them, it is necessary to formally define the notion of concept that is used here.

Following Frege, a concept is defined as a function $f : D \rightarrow \{\bot; \top\}$, where D is a domain of objects and $\{\bot; \top\}$ truth values. Concepts are associated to unsaturated entities and objects to saturated entities. (e.g. the concept is-HUMAN( ) can be applied to the saturated object John to return true).

Entities of the ontology of time and aspect (intervals) are referred to as types[8], and concepts and types are related by the following equivalence[9] :

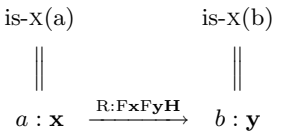
$$a : \boldsymbol{x} \quad \text{iff} \quad \text{is-X(a)} = \top \ . \tag{3}$$

The left clause of this equivalence can also be expressed by the proposition *a is an instance of the type* $\boldsymbol{x}$. The definition of a concept is given by writing (e.g. for a type **x**), is-X(a) ≡ "*a has the properties inherited from the concept is-X()*". Whence if an object has a given type, it inherits all properties associated to the concept (e.g. John:**human** ≡ is-HUMAN(John) → has-body(John) and has-mind(John)...).

A relation (here binary) has a functional type built recursively using the rule (1). In the following diagram, R is the name of an arbitrary binary relation holding between two typed objects.

[8] The closest notion of type used in this sense can be found in the literature on ontologies in computer science under the term universal (taken from philosophy) (See for instance [13]). Types are used to express general relations that hold between classes of objects, like types in categorial grammars or the TBox level in description logics.

[9] Types are written with bold lower-case letters, and name of concepts are written with upper-case.



$$\dfrac{\dfrac{R : F\mathbf{x}F\mathbf{y}\mathbf{H} \qquad a : \mathbf{x}}{(Ra) : F\mathbf{y}\mathbf{H}} \qquad b : \mathbf{y}}{((Ra)b) : \mathbf{H}}\ .$$

Fig. 2. The same binary relation R is represented graphically (*on the left*) and by its applicative tree (*on the right*). This tree rests on the application rule (2). The type H corresponds to truth values. Double lines in graphical representation refer to the equivalence (3).

3.2 Definitions of Concepts and Relations in the Ontology of Time and Aspect

As it has been mentioned, linguistic entities identified in section 2.1, namely, the different topological intervals (of instants) and temporal reference systems are identify to primitive types in the formal ontology. Other types are introduced for technical reasons that are explained below.

Table 1. This table establishes the typology of entities required in the formal ontology of time and aspect

Types	Entity description
H	*Truth values*
ref	*Temporal reference system (see section 2.1)*
inst	*Instant*
intv	*Interval*
intv-topo	*Interval with topological properties*
intv-topo-B$^-$	*Unbounded interval*
intv-topo-B$^+$	*Bounded interval*
intv-topo-B$^+$-cl	*Closed bounded interval (see section 2.1)*
intv-topo-B$^+$-op	*Open bounded interval (see section 2.1)*
intv-topo-B$^+$-ho	*Half-open bounded interval (see section 2.1)*

Remark 1. Having established three different types respectively for closed, open or half-open intervals, it is possible to express polymorphic relations between specific intervals. For instance, a relation R holding between a closed and a half-open interval will have the functional type R: F **intv-topo-B$^+$-cl** F **intv-topo-B$^+$-op H**. All arguments with other types than those in the signature of the relation[10] will lead to a type error.

[10] This means is used to express semantic constraints between aspectual values. For instance, two events (respectively true on closed intervals) cant be adjacent (or "meet" in Allen´ s terminology or also share a bound) according to theory introduced in section 2.1 (there is necessarily a state in between).

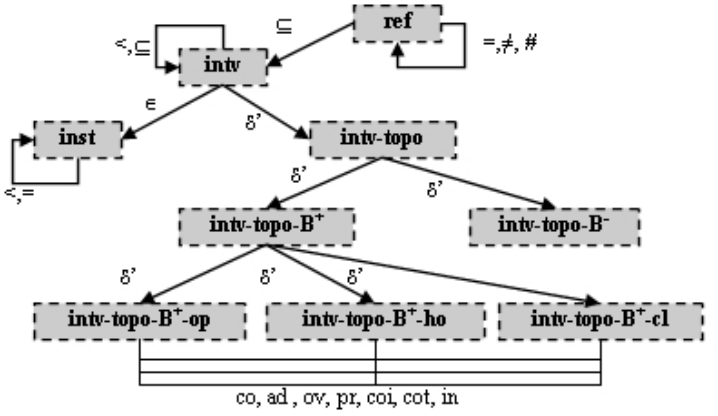


Fig. 3. Each relation in this ontology is typed. For instance the relation "=" can have the type F **inst** F **inst H** or F **ref** F **ref H**. The seven relations (co, ad, ov, pr, coi, cot, in) are defined in table 2. The relation δ' is defined from the determination operator δ (see [14]), and shares some properties with the subsumption relation.

An ontology being not a simple typology (unstructured set of entities), the next figure graphically represents specific articulations of types taken from table 1, and then a conceptual content is given (definitions following the figure 3) to types (following the equivalence (3)) and to relations.

Definition 1. *A temporal reference system (type **ref**) is a strict total ordered set $(T, <)$ where T is a non-empty set of instants. "$<$" is called the precedence relation and verifies the additional properties of density and continuous cut (see footnote 6).*

$$is\text{-REF}(R) \equiv R = (T, <) \wedge T \neq \varnothing \tag{4}$$

Definition 2. *An interval (type **intv**) is a non-empty convex subset of a temporal reference system.*

$$\begin{aligned} is\text{-INTV}(I) \equiv \exists R(is\text{-REF}(R)) \wedge I \neq \varnothing \\ \wedge\ \forall a, b \in I(a < b, \forall t \in R(a \leq t \leq \Rightarrow t \in I)) \end{aligned} \tag{5}$$

Definition 3. *A topological interval (type **intv-topo**) is an interval to which operators of the point-set topology like* interior, boundary *or* closure *can be applied*[11].

[11] Indeed, open intervals of any totally ordered set can define a topology on this set. Here, there exists a topological space (T, O) where T is the non-empty set of instants and O is a topology on T consisting of all open subsets of T verifying the specific topological axioms.

Here are recalled the basic topological notions from which more specific topological intervals will be defined like the closed or half-open intervals.

Given an interval I and a temporal reference system R such that $I \subseteq R$.

- The *interior* of I, denoted $Int(I)$, is defined by the union of all open intervals include in I. If I is open then $Int(I) = I$.
- The *closure* of I, denoted $Cl(I)$, is defined by the intersection of all closed intervals including I. If I is closed then $Cl(I) = I$.
- The *boundary* of I, denoted $Bd(I)$, is defined by the intersection of the closure of I and the closure of the complement of I.
- It is possible to add for an interval I of totally ordered instants, its *right bound*, denoted $BdR(I)$, defined by $BdR(I) = max(Bd(I))$, and its *left bound*, denoted $BdL(I)$, defined by $BdL(I) = min(Bd(I))$.

Definition 4. *Respectively, unbounded topological interval (type* ***intv-topo-B$^-$****) and bounded topological interval (type* ***intv-topo-B$^+$****) are defined by,*

$$is\text{-INTV-TOPO-B}^- (I) \equiv is\text{-INTV-TOPO}(I) \wedge (BdR(I) = \infty \vee BdL(I) = \infty) \quad (6)$$

$$is\text{-INTV-TOPO-B}^+ (I) \equiv is\text{-INTV-TOPO}(I) \wedge (BdR(I) \neq \infty \wedge BdL(I) \neq \infty) \quad (7)$$

Definition 5. *Respectively, closed interval (type intv-topo-B$^+$-cl), open interval (type intv-topo-B$^+$-op) and half-open (at right) interval (type intv-topo-B$^+$-ho) are defined by,*

$$is\text{-INTV-TOPO-B}^+\text{-CL(I)} \equiv is\text{-INTV-TOPO-B}^+(\text{I}) \wedge I = Cl(I) \quad (8)$$

$$is\text{-INTV-TOPO-B}^+\text{-OP(I)} \equiv is\text{-INTV-TOPO-B}^+(\text{I}) \wedge I = Int(I) \quad (9)$$

$$is\text{-INTV-TOPO-B}^+\text{-HO(I)} \equiv is\text{-INTV-TOPO-B}^+(\text{I}) \wedge (I = Int(I) \cup BdL(I)) \quad (10)$$

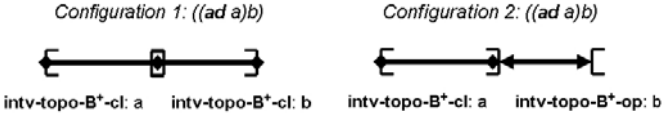
Concepts being defined, the next paragraph will focus on relations over topological intervals defined in defintion 5 (closed, open and half-open). They can be defined from "<": F **inst** F **inst H** and $\subseteq$: F **intv** F **intv H** being respectively the precedence relation between instants and the classical set-theoretic inclusion between intervals.

The difference with Allen relations ([15]) or Van Benthem[12] definitions is that the semantics of bounds is taken into consideration and related to a logico-linguistic analysis. Each relation is provided with a set of admissible types for its arguments called its signature, and as mentioned in remark 1, this signature is used to avoid certain undesired configurations between topological intervals (semantic constraints of the ontology, see figure 1). For instance, given the relation "ad" with the signature {F **intv-topo-B$^+$-cl** F **intv-topo-B$^+$-op H**}, and the following configurations.

[12] Here, definitions are based on "<" and "$\subseteq$" as in period structures from [16], but periods are defined differently.

Table 2. This table provides definitions for relations holding between topological intervals

Symb.	Name	Definition	Rep.
co	coincidence	$A\ co\ B \equiv A \subseteq B \wedge B \subseteq A$	
ad	adjacence	$A\ ad\ B \equiv BdR(A) = BdL(B)$	
ov	overlap	$A\ ov\ B \equiv \exists i(i \subseteq A \wedge i \subseteq B) \wedge BdL(A) < BdL(B)$	
pr	precedence	$A\ pr\ B \equiv BdR(A) < BdL(B)$	
coi	initial coincidence	$A\ coi\ B \equiv A \subset B \wedge BdL(a) = BdL(B)$	
cot	terminal coincidence	$A\ cot\ B \equiv A \subset B \wedge BdR(a) = BdR(B)$	
in	interiority	$A\ in\ B \equiv BdL(B) < BdL(A) \wedge BdR(B) < BdR(A)$	



Configuration 1 will lead to a semantic type error[13] because the type "F **intv-topo-B$^+$-cl** F **intv-topo-B$^+$-cl H**" is not included in the signature of the relation "ad" whereas configuration 2 is well-typed (e.g. value of resulting state, see figure 1).

This ontology with specific semantic constraints being developed, it enables the definition of a specific model $\langle I, R, V\rangle$ where,

1. I is the set of all open, closed or half-open intervals defined by $I = \{\forall i; \text{is-INTV-TOPO-B}^+\text{-HO}(i) \vee \text{is-INTV-TOPO-B}^+\text{-OP}(i) \vee \text{is-INTV-TOPO-B}^+\text{-CL}(i)\}$
2. R={co, ad, ov, pr, coi, cot, in} the set of typed binary relations over I.
3. $V : PR \rightarrow \wp(I)$ a valuation function assigning to each predicative relation in the set PR a subset of I where it is realized .

4 Conclusion

The main contribution of the article lies in the establishment of the formal ontology of time and aspect (section 3) as a means or a toolkit to express formally some specific semantic constraints analyzed from aspectual situations in texts.

References

1. Desclés, J.-P.: Construction formelle de la catégorie grammaticale de l'aspect, Paris, Klincksieck, Metz, pp. 195–237 (1980)
2. Comrie, B.: Aspect: An Introduction to the Study of Verbal Aspect and Related Problems. Cambridge University Press, Cambridge (1976)

[13] The meeting point being contained in both intervals, a proposition could have contradictory truth value at this point. (e.g. *to be standing/to be sitting down*)

3. Vendler, Z.: Verbs and times. The philosophical review 66(2), 143–160 (1957)
4. Dowty, D.R.: Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's Ptq, October 1979. D. Reidel Publishing Company, Dordrechtz (1979)
5. Verkuyl, H.J.: Aspectual classes and aspectual composition. Linguistics and philosophy 12(1), 39–94 (1989)
6. Mourelatos, A.P.D.: Events, processes, and states. Linguistics and philosophy 2(3), 415–434 (1978)
7. Krifka, M.: Thematic relations as links between nominal reference and temporal constitution. Lexical matters 24 (1992)
8. Pustejovsky, J.: The syntax of event structure. Cognition 41(1-3), 47–81 (1991)
9. Desclés, J.-P.: Combinatory logic, language and cognitive representations. In: Weingartner, P. (ed.) Alternative Logics. Do Sciences Need them?, pp. 115–148. Springer, Heidelberg (2005)
10. Desclés, J.P., Guentcheva, Z.: Is the notion of process necessary? In: Bertinetto, M. (ed.) Temporal Reference Aspect and Actionality, Turin, pp. 55–70 (1994)
11. Montanari, A.: Back to interval temporal logics. In: Garcia de la Banda, M., Pontelli, E. (eds.) ICLP 2008. LNCS, vol. 5366, pp. 11–13. Springer, Heidelberg (2008)
12. Guarino, N., Giaretta, P.: Ontologies and knowledge bases - towards a terminological clarification. In: Mars, N. (ed.) Towards Very Large Knowledge Bases, pp. 25–32. IOS Press, Amsterdam (1995)
13. Bittner, T., Donnelly, M., Smith, B.: Individuals, universals, collections: On the foundational relations of ontology. In: Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS), pp. 37–48 (2004)
14. Desclés, J.-P., Pascu, A.: Logic of determination of object a formal concept analysis (2007)
15. Allen, J.F.: Maintaining knowledge about temporal intervals. ACM Commun. 26(11), 832–843 (1983)
16. Van Benthem, J.: The Logic of Time: A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse, 2nd sub edn., March 1991. Springer, Heidelberg (1991)
17. Loebe, F., Herre, H.: Formal semantics and OntologiesTowards an ontological account of formal semantics. In: Proceeding of the 2008 conference on Formal Ontology in Information Systems, pp. 49–62. IOS Press, Amsterdam (2008)
18. Cohn, A.: Taxonomic reasoning with many-sorted logics. Artificial Intelligence Review 3(2-3), 89–128 (1989)
19. Desclés, J.-P., Vanderveken, D.: Reasoning and Aspectual-Temporal calculus. In: Logic, Thought and Action, vol. 2, pp. 217–244. Springer, Heidelberg (2005) (Neitherlands edn.)

A Non-linear Semantic Mapping Technique for Cross-Language Sentence Matching

Rafael E. Banchs and Marta R. Costa-jussà

Barcelona Media Innovation Centre, Barcelona, Spain
{rafael.banchs,marta.ruiz}@barcelonamedia.org

Abstract. A non-linear semantic mapping procedure is implemented for cross-language text matching at the sentence level. The method relies on a non-linear space reduction technique which is used for constructing semantic embeddings of multilingual sentence collections. In the proposed method, an independent embedding is constructed for each language in the multilingual collection and the similarities among the resulting semantic representations are used for cross-language matching. It is shown that the proposed method outperforms other conventional cross-language information retrieval methods.

Keywords: Cross-language Information Retrieval, Semantic Mapping, Multi-dimensional Scaling.

1 Introduction

Cross-language information retrieval (CLIR), which is a subfield of the traditional information retrieval (IR), provides users with access to information that is in a different language from their queries. CLIR is gaining more and more attention as the availability of information in languages different from English increases in the Internet. It has become one popular research area in information retrieval during the last 10+ years [1]. Research in CLIR has been significantly encouraged by three well-known evaluation campaigns: a cross-language information retrieval track at TREC, the Cross-Language Evaluation Forum (CLEF) and the NTCIR Asian Language Evaluation. Recently, some CLIR real applications have appeared such as the cross-language search by Google on 2007 and the user-driven multilingual news aggregation Europe Media Monitor.

Given a query in a given source language, the aim of CLIR is retrieving relevant documents in a target language. In [2], four different strategies for matching a query with a set of documents in the context of CLIR were identified: cognate matching, document translation, query translation and interlinguas. Nowadays, one of the most popular approaches is query translation. However, this approach is bilingual in nature and in a highly multilingual environment with, for instance, *N* languages, it may be impractical as *N*(N-1)* translation directions must be accounted for. On the contrary, an interlingua-based approach would only require *N* mappings or translations to be accounted for. In this sense, this latter strategy seems to be more suitable in those applications involving a large number of languages.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 57–66, 2010.

In this work, we focus on the specific problem of cross-language text matching at the sentence level. In this problem, a segment of text in a given source language is used as query for recovering a similar or equivalent segment of text in a different target language. This task is essential to some specific applications such as parallel corpora compilation [3] and cross-language plagiarism detection [4].

We address the problem under consideration by means of an interlingua-based CLIR system that follows a non-linear semantic mapping approach similar to the one presented in [5]. Semantic mapping techniques have been successfully used for concept association and related-term identification tasks [6], [7]. We illustrate here, that this kind of non-linear mapping can constitute a very effective and valuable strategy for the problem under consideration. Some other recent approaches have achieved interesting results in CLIR applications by using regression canonical correlation analysis (an extension of canonical correlation analysis) [8].

The rest of the paper is structured as follows. In section 2, the implemented interlingua-based CLIR method is described. In section 3, the proposed methodology is illustrated by performing cross-language text matching at the sentence level on a penta-lingual document collection. Also, within this section, the performance quality of the implemented system is evaluated and compared against two conventional CLIR systems, showing that the proposed approach outperforms the other two. Finally, in section 4, the most relevant conclusions derived from the experimental results are presented.

2 Cross-Language Semantic Mapping

The fundamental issue behind the proposed CLIR method is the idea of semantic mapping. As illustrated in [5], starting from the term-document matrix representation of a given document collection, it is possible to build a semantic representation for the collection by using the non-linear projection technique known as multidimensional scaling [9]. Moreover, if a multilingual parallel document collection is available, a semantic map can be computed independently for each language's document subset, and the resulting maps will exhibit a high degree of similarity among them. The observed similarities among the maps are mainly because such maps are able to capture the most prominent semantic relationships among the documents within the collection, which are, indeed, language independent.

The structural similarities observed among the different semantic maps provide the possibility of placing documents from different languages into any other language generated map. In this way, these maps can be actually interpreted as an interlingua type of representation. The general procedure for CLIR by means of semantic mapping can be summarized as follows:

- start from a multilingual collection of "anchor documents" and construct the retrieval-language semantic map,
- place new documents and queries from any source language into the retrieval map by using a linear transformation matrix, and
- retrieve documents by using a distance or similarity metric.

Figure 1 provides a schematic representation of interlingua-based CLIR by means of non-linear semantic mapping.

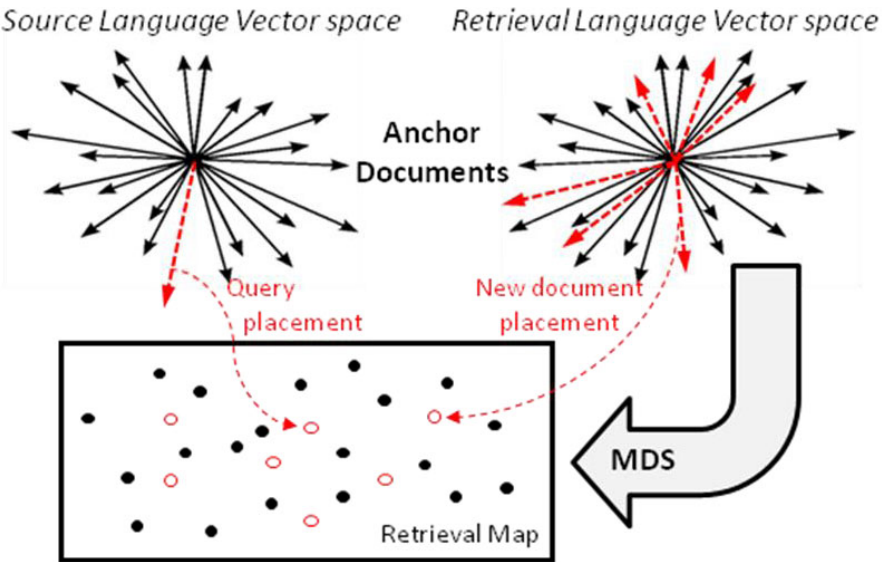


Fig. 1. Schematic representation of interlingua-based CLIR by means of semantic mapping

A linear transformation operator T for projecting documents or queries from the original high-dimensional space into the designated low-dimensional semantic map can be inferred from the multilingual set of anchor documents as follows:

$$M = T\,D \;\Rightarrow\; T = M\,D^{-1} \tag{1}$$

where D is a square matrix of size NxN (being N the number of available anchor documents) which contains the distances among the anchor documents in the original high-dimensional space (the document similarity matrix), and M is the KxN matrix containing the coordinates of the anchor documents in the reduced k-dimensional semantic map.

The matrix M, which represents the coordinates for anchor documents in the computed semantic map, is obtained by applying MDS to similarity matrix D. More specifically, the algorithmic setting for the proposed methodology considers using the cosine distance for constructing the similarity matrix D and Sammon's projection criterion for computing the semantic map M [10].

Different from the procedure described in [5], where a "monolingual" projection operator was computed, here we compute a "cross-language" projection operator, for which M is computed on the retrieval language and D is computed on the source language. This "cross-language" variant of the method has been proven to provide better results than the original "monolingual" projection operator [11].

Once the projection operator has been computed, any new probe document or query can be placed into the designated retrieval map by using:

$$m = T\,d \tag{2}$$

where *d* represents a vector containing the distances between the probe document (or query) and the anchor documents in the original high-dimensional space, *T* is the projection operator defined in (1), and *m* is a vector containing the coordinates for the probe document (or query) in the low-dimensional map.

Additionally, as many maps can be generated as there are different languages in the multilingual collection, we propose a multiple map combination approach based on a majority voting strategy. According to this, a retrieval map is constructed for each language in the multilingual collection. Then, all probe documents and queries are projected into all maps, where similarities are computed and individual rankings are performed. Finally, a global ranking is obtained by majority voting of all individual rankings. This procedure is illustrated in Figure 2.

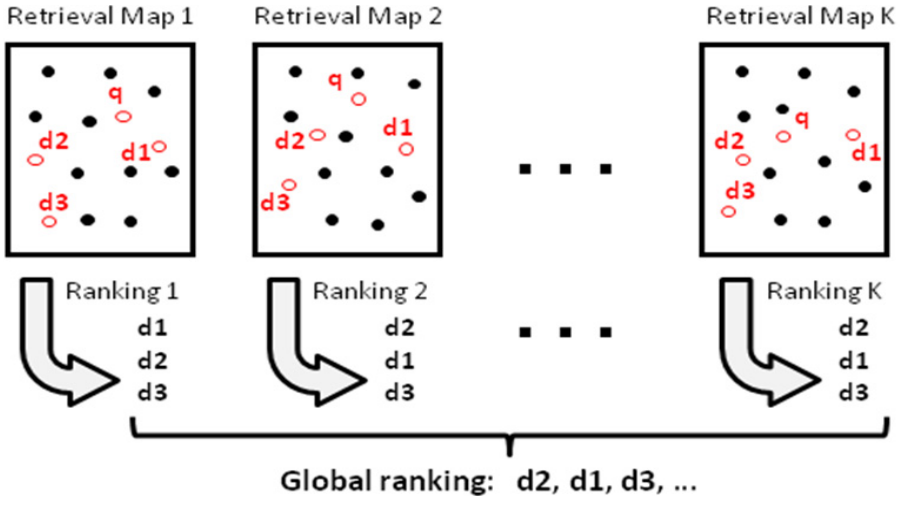


Fig. 2. Majority voting strategy implemented for combining individual map rankings

3 Cross-Language Sentence Matching Experiments

As already mentioned in the introduction, in this work, we focus on the problem of cross-language text matching at the sentence level. In this particular task, a segment of text in a given source language is used as query for recovering an equivalent segment of text in a different target language. In this section, the methodology described above is evaluated and compared with other two CLIR approaches: another interlingua-based approach which is based on latent semantic indexing [12], and a more conventional query translation approach [13] which considers a cascade combination of a machine translation system and a monolingual IR system.

3.1 Multilingual Sentence Dataset

The dataset considered for the experiments is a multilingual sentence collection that was extracted from the Spanish Constitution, which is available for downloading at

the Spanish government's main web portal: www.la-moncloa.es. In this website, all constitutional texts are available in five different languages, including the four official languages of Spain: Spanish, Català, Galego and Euskera, as well as English.

The texts are organized in 169 articles plus some additional regulatory dispositions. All texts were segmented into sentences and the resulting collection was filtered according to sentence length. More specifically, sentences having less than five words were discarded aiming at eliminating titles and some other non-relevant information. The resulting multilingual sentence collection was randomized and a test set of 200 sentences was extracted.

Table 1 summarizes the main statistics for both the overall collection and the selected test subset.

Table 1. Main statistics for the overall multilingual dataset and the selected test set

Overall Dataset	English	Spanish	Català	Euskera	Galego
Number of sentences	611	611	611	611	611
Number of words	15285	14807	15423	10483	13760
Vocabulary size	2080	2516	2523	3633	2667
Average sentence length	25.01	24.23	25.24	17.16	22.52
Selected Test Set	English	Spanish	Català	Euskera	Galego
Number of sentences	200	200	200	200	200
Number of words	4667	4492	4669	3163	4175
Vocabulary size	1136	1256	1273	1618	1316
Average sentence length	23.34	22.46	23.34	15.82	20.88

Finally, and for illustrative purposes, one sample sentence from the multilingual collection is presented in Table 2.

Table 2. A sample sentence from Spanish Constitution's multilingual sentence collection

Language	**Sample sentence**
English	This right may not be restricted for political or ideological reasons.
Spanish	Este derecho no podrá ser limitado por motivos políticos o ideológicos.
Català	Aquest dret no podrà ser limitat per motius polítics o ideològics.
Euskera	Eskubide hau arrazoi politiko edo idiologikoek ezin dute mugatu.
Galego	Este dereito non poderá ser limitado por motivos políticos ou ideolóxicos.

3.2 Experimental Evaluation of the Proposed Technique

In this subsection, the proposed methodology is illustrated by performing cross-language sentence matching on the Spanish Constitution's multilingual collection, and its performance quality is evaluated by means of the top-1 and top-5 accuracies measured over the test subset that was described in Table 1. The specific task to be considered consists of recovering a sentence from the English version of the Spanish Constitution using as a query the same sentence in any of the four Spanish languages: Spanish, Català, Euskera and Galego.

For constructing the CLIR system, four hundred sentences were randomly selected from the remaining portion of the dataset that did not include the test set. This subset

of four hundred sentences constituted the anchor document collection that was used for constructing the maps. One map was constructed for each of the five languages available in the collection by using multidimensional scaling.

The space dimension of the constructed maps was set to 350. As already reported in [5] and [11], where experiments considering a full range of reductions were presented, space reductions down to dimensionalities above 75% the size of the anchor document collection provide appropriate embeddings for MDS- and LSI-based methods to be comparable. Notice also that reducing the dimensionality down to 350 implied overall space reductions ranging from 83% (in the case of English) to 90% (in the case of Euskera).

Finally, following (1) and (2), transformation matrices were constructed for all constructed semantic maps and all test sentences from each language were placed into them. Sentence matching was performed at each individual map by using the cosine distance as a similarity metric.

Table 3 summarizes results for all sentence matching exercises conducted over the five constructed maps, as well as the implemented majority voting strategy, and the four considered query languages: Spanish, Català, Euskera and Galego. For instance, results reported in row Galego and column Català correspond to sentence matching conducted between Català (query language) and English (target language) over the semantic map constructed from Galego anchor documents.

Table 3. Top-1 and Top-5 accuracies for all conducted experiments on cross-language sentence matching based on semantic maps

	Spanish		Català		Euskera		Galego	
Retrieval Map	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
English	97.0	**100.0**	96.0	99.0	69.5	91.0	**95.0**	98.5
Spanish	95.5	99.0	94.5	99.5	**77.0**	**93.0**	94.0	**99.5**
Català	95.0	**100.0**	94.5	99.5	74.5	90.5	93.0	99.0
Euskera	96.5	99.0	95.0	99.5	70.0	86.5	**95.0**	98.5
Galego	96.5	**100.0**	94.5	**100.0**	73.0	91.5	93.0	98.0
Majority voting	**97.5**	**100.0**	**96.5**	99.5	76.0	92.5	94.5	**99.5**

Several interesting observations can be derived from results reported in Table 3. First of all, it can be seen that regardless the semantic map used for sentence matching the best scores are always obtained when Spanish is the query language. This might be explained by the fact that the constitutional texts used are originally Spanish texts, which have been further translated into the other four languages. According to this, it could be expected for Spanish sentence projections to be more coherent than other language projections across all maps; and, in consequence, it would be reasonable to assume that best scores should be obtained in those cases where Spanish is either the query or the retrieval language.

On the other hand, the worst results are consistently obtained for all cases in which Euskera is the query language. This is explained by the morphological complexity of the language, which is evidenced in Table 1 as it exhibits the largest vocabulary and the smallest number of running words. Nevertheless, surprisingly, when the retrieval semantic map is derived from the Euskera anchor documents, resulting scores are as

good as the ones obtained from any other of the maps. This verifies the high degree of language independence the generated semantic maps can provide.

Another interesting observation that can be derived from Table 3 is the fact that, with the exception of Euskera queries, the English map is the best single semantic map for sentence matching when considering top-1 matches. At a first glance, one may think that this must be related to English being the target language of the task under consideration. Nevertheless, if top-5 matches are considered, best results are achieved with semantic maps constructed from Galego and Spanish anchor document collections, which does not support the previous finding as well as does not seem to have any logical justification. In this sense, further research will be needed to come up with a clear understanding on these issues.

Finally, it can be also concluded from Table 3 that the majority voting strategy implemented for combining all semantic maps is not actually providing a significant improvement on the sentence matching task under consideration. Indeed, it is only in two cases (Spanish top-1 and Català top-1) that majority voting is providing an actual improvement. In all other cases, majority voting results are equal to or less than the best single map result. This clearly suggests that majority voting over single map rankings might not be the best strategy to follow. According to this, further research will be needed to determine the best map combination strategy, which might include, for example, some optimization procedure over a linear combination of the sentence similarities computed on the different maps.

3.3 Comparative Evaluation of the Proposed Technique

In this subsection, the proposed methodology is compared with two other referential CLIR systems: the LSI-based approach proposed in [12] and the more commonly used query translation approach [13]. Similar to the previous experiments, the task consists of recovering an English sentence using the same sentence in any of the other four languages as a query, and the performance quality is evaluated in terms of the top-1 and top-5 accuracies over the same test subset described in Table 1.

The first contrastive system to be considered implements the LSI-based CLIR technique described in [12]. This system applies single value decomposition (SVD) to a concatenated matrix of monolingual vector space representations generated from a multilingual document collection. This generates a low dimensional interlingua vector space representation. The main difference between this procedure and the non-linear semantic mapping method proposed in this work is the linear nature of the singular value decomposition algorithm. In order for the results to be comparable, the same subset of four hundred anchor documents (previously used for constructing the MDS-based semantic maps) were used for constructing the LSI-based vector space representations, for which reduced space dimensionality was also set to 350. One LSI-based vector space representation was computed for each of the four considered query-retrieval language pairs.

The second contrastive system to be considered implements a query translation strategy followed by a standard monolingual information retrieval approach. For the query translation step, the Opentrad platform was used `http://www.opentrad.com/` [14]. This constitutes a state-of-the-art machine translation service that provides automatic translation among several language pairs including the

four Spanish languages plus English, Portuguese and French. However, it must be advised that as the Euskera-to-English translation direction is not provided by this service, the considered task could not be evaluated with this contrastive system for this specific language pair. On the other hand, the monolingual information retrieval step was implemented by using Solr, which is an XML-based open-source search server based on the Apache-Lucene search library [15].

Table 4, summarizes the results obtained from the comparative evaluation between the proposed semantic map based methodology and the two contrastive systems. Only those results corresponding to the majority voting strategy are reported for semantic maps (for comparing these results to individual semantic mapping results, the reader can refer to Table 3).

Table 4. Comparative evaluation of the proposed method (majority voting of semantic maps) vs. LSI-based and query translation CLIR techniques

	Spanish		Català		Euskera		Galego	
CLIR Method	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
LSI based	96.0	99.0	95.5	98.5	75.5	90.5	93.5	97.5
Query transl.	96.0	99.0	95.5	99.0	-	-	93.5	98.0
Semantic maps	**97.5**	**100.0**	**96.5**	**99.5**	**76.0**	**92.5**	**94.5**	**99.5**

As seen from the table, the proposed methodology clearly outperforms the two contrastive systems in all the cases. Notice, however, that the observed differences are small as all systems are providing high accuracy values in most of the cases. Some additional experimentation has been conducted with several different subsets of the same 200-sentence test dataset, and similar results have been consistently obtained. This suggests that the observed differences among the methods in Table 4, although small, are significant.

The results reported in Table 4 show that non-linear semantic maps described here seem to be more suitable for cross-language sentence matching than both: the linear projections provided by the LSI-based method, and the language conversions provided by state-of-the-art machine translation. Also, it can be verified that both contrastive systems perform the same for top-1 accuracies, but query translation outperforms the LSI-based approach for top-5 accuracies.

4 Conclusions and Future Work

A non-linear semantic mapping procedure has been implemented for cross-language text matching at the sentence level. The proposed method relies on a non-linear space reduction technique (multidimensional scaling) for constructing semantic embeddings of a given multilingual document collection. These semantic representations can be exploited for implementing an interlingua-based CLIR system.

In the considered CLIR task, a segment of text in a given source language is used as query for recovering a similar or equivalent segment of text in a different target language. The proposed method is evaluated and compared against two conventional cross-language information retrieval methods over a penta-lingual sentence collection

extracted from the Spanish Constitution. Results presented in this work show that the proposed methodology outperforms the other two methods on the specific task under consideration.

Despite the positive results, the majority voting strategy that was implemented for combining the individual rankings obtained from different semantic maps does not seem to provide any significant improvement with respect to the independent use of the individual semantic maps. In this sense, further research will be needed to determine the best combination strategy, which might include, for example, some optimization procedure over a linear combination of the sentence similarities computed on the different maps.

Additionally, some other interesting problems that must be addressed in future research have been also identified:

- To evaluate the performance of the proposed method under specific scenarios in which comparable corpora, instead of parallel corpora, are considered.
- To test the method in more realistic settings in which the queries, the documents to be retrieved and the anchor documents used for constructing the semantic maps are not necessarily in the same domain.
- To study and evaluate possible issues related to performance and scalability when multilingual datasets much larger than the one used in this work are involved.
- To design and evaluate methods for replacing current linear projection matrices by non-linear transformation operators for probe document and query placement.
- To study in more detail any possible relationship between the performance of the system and the languages involved: the query language, the target language and the language used for constructing the retrieval map.
- To evaluate the performance of the proposed methodology in other different CLIR tasks.

Acknowledgments. This research has been partially supported by the Spanish Government's *Juan de la Cierva* program and the BUCEADOR project (TEC2009-14094-C04-01). Additionally, the authors would like to thank Barcelona Media Innovation Centre for its support and permission to publish this work.

References

1. Kishida, K.: Technical issues of cross-language information retrieval: a review. Information Processing and Management 41(3), 433–455 (2005)
2. Oard, D.W., Diekema, A.R.: Cross-language information retrieval. Annual Review of Information Science Technology (ARIST) 33, 223–256 (1998)
3. Utiyama, M., Tanimura, M.: Automatic construction technology for parallel corpora. Journal of the National Institute of Information and Communications Technology 54(3), 25–31 (2007)

4. Potthast, M., Stein, B., Eiselt, A., Barrón, A., Rosso, P.: Overview of the 1st international competition on plagiarism detection. In: Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (2009), http://ceur-ws.org/Vol-502
5. Banchs, R., Kaltenbrunner, A.: Exploiting MDS projections for cross-language information retrieval. In: 31st Annual International ACM SIGIR Conference, pp. 863–864 (2008)
6. van Eck, N., Waltman, L., van den Berg, J.: A novel algorithm for visualizing concept associations. In: 16th International Workshop on Database and Expert System Applications, pp. 405–409 (2005)
7. Banchs, R.: Semantic mapping for related term identification. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 111–124. Springer, Heidelberg (2009)
8. Rupnik, J., Shawe-Taylor, J.: Multiview canonical correlation analysis and cross-lingual information retrieval (2008), http://videolectures.net/lms08_rupnik_rcca/
9. Cox, M.F., Cox, M.A.: Multidimensional Scaling. Chapman & Hall, UK (2001)
10. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Transactions on Computers 18, 401–409 (1969)
11. Banchs, R., Costa-jussà, M.: Extracción crosslingüe de documentos usando mapas semánticos no lineales. Procesamiento del Lenguaje Natural 43, 169–176 (2009)
12. Dumais, S., Landauer, T., Littman, M.: Automatic cross-linguistic information retrieval using latent semantic indexing. In: SIGIR 1996 Workshop on Cross-Lingual Information Retrieval (1996)
13. Chen, J., Bao, Y.: Cross-language search: the case of Google language tools. First Monday 14(3-2) (2009)
14. Ramírez, G., Sánchez, F., Ortiz, S., Pérez, J., Forcada, M.: Opentrad Apertium open-source machine translation system: an opportunity for business and research. In: 28th Conference on Translating and the Computer (2006)
15. The Apache Solr Tutorial, http://lucene.apache.org/solr/tutorial.html

Comparison of Paraphrase Acquisition Techniques on Sentential Paraphrases

Houda Bouamor, Aurélien Max, and Anne Vilnat

LIMSI-CNRS & Univ. Paris-Sud
F-91403 Orsay, France

Abstract. In this article, the task of acquisition of subsentential paraphrases is discussed and several automatic techniques are presented. We describe an evaluation methodology to compare these techniques and some of their combinations. This methodology is applied on two corpora of sentential paraphrases obtained by multiple translations. The conclusions that are drawn can be used to guide future work for improving existing techniques.

Keywords: Paraphrase, Monolingual bi-phrase patterns.

1 Introduction

Deciding whether two text units convey the same meaning is one of the most important needs in Natural Language Processing. As natural language offers many possible alternatives for expression, the ability to determine that two words or phrases have equivalent meaning in context is required for analyzing text. In question answering, for instance, this can be used to extract correct answers expressed with words that differ from those in the question. Large-scale acquisition of sets of equivalent text units is an active field of research. Applications can be in text analysis, for example to allow different wordings in Machine Translation evaluation [9], or in text generation, for example to help writers to find more appropriate wordings [14].

A number of techniques have been proposed for acquiring text units in a *paraphrasing* relationship, defined by a reciprocal textual entailment between the two units. These techniques are designed to acquire paraphrases from specific types of resources. Monolingual corpora have been extensively used to test the *distributional hypothesis*, which states that text units occurring in similar contexts may be paraphrases. The limitation in themes or genres in a comparable corpus increases the probability of extracting accurate paraphrase pairs in context. Bilingual parallel corpora have also been used to test the translation equivalence hypothesis, which states that text units sharing translations in at least one other language may be paraphrases.

In contrast, few works have addressed using *monolingual parallel corpora* made up of paraphrases at the sentential level. This can be explained by the facts that very few such corpora are available and that their construction can be

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 67–78, 2010.

costly and difficult to design. However, because they associate sentences which express the same meaning, such corpora allows for the most reliable acquisition of paraphrase pairs. Contrary to what is the case with comparable monolingual corpora or parallel bilingual corpora, paraphrases can be observed directly, and examples of contexts in which one may substitute one with the other can be extracted straightforwardly.

This work focusses on the acquisition of accurate paraphrases in context from monolingual parallel corpora, and on combining the results obtained from different techniques. The remainder of the paper is organised as follows. In section 2 we will review the main approaches for acquiring subsentential paraphrases and then describe three particular existing techniques that can be applied on sentential paraphrases in section 3: a technique based on statistical learning of word alignments, one based on the symbolic representation of linguistic variation, and another based on the syntactic fusion of sentences. An experimental framework for comparing and combining the outputs of these techniques will be described in section 4. Our methodology for building a suitable corpus by multiple translation will be explained as well as an existing methodology for evaluating the performance of the various techniques. Lastly, we will conclude and describe our future work in section 5.

2 Previous Work on Subsentential Paraphrase Acquisition

The hypothesis that if two words or, by extension, two phrases, occur in similar contexts then they may be interchangeable has been extensively tested. This *distributional hypothesis*, attributed to Zellig Harris, was for example applied to syntactic dependency paths in the work of Lin and Pantel [13]. Their results take the form of equivalence patterns with two arguments such as {*X asks for Y*, *X requests Y*, *X's request for Y*, *X wants Y*, *Y is requested by X*, ...}.

Using comparable corpora, where the same information probably exists under various linguistic forms, increases the likelihood of finding very close contexts for subsentential units. Barzilay and Lee [2] propose a multi-sequence alignment algorithm that take structurally similar sentences and build a compact lattice representation that encode local variations. The work by Bhagat and Ravichandran [4] describes an application of a similar technique on a very large scale.

The hypothesis that two words or phrases are interchangeable if they share a common translation into one or more other languages has also been extensively followed in works on subsentential paraphrase acquisition. Bannard and Callison-Burch [1] describe a pivoting approach that can exploit bilingual parallel corpora in several languages. The same technique has been applied to the acquisition of local paraphrasing patterns in Zhao *et al.* [17]. The works of Callison-Burch [5] and Max [14] have shown how the monolingual context of a sentence to paraphrase can be used to improve the quality of the acquired paraphrases.

Another approach consists in modelling local paraphrasing identification rules. The work of Jacquemin on the identification of term variants [8], which exploits

rewriting morphosyntactic rules and descriptions of morphological and semantic lexical families, can be extended to extract the various forms corresponding to input patterns from large monolingual corpora.

All the previous approaches can produce inappropriate pairs that do not correspond to paraphrastic variants in context. This is largely due to the fact that the corpora that they use never explicitly encode paraphrasing relationships between text units. For instance, a paraphrase obtained by pivoting through another language may not have been observed in the context of the original phrase: the phrase *it is too early to* could be automatically extracted for the original phrase *this is not the time to* by pivoting through the French phrase *il est trop tôt pour*, an acceptable translation for some occurrences of the two English phrases, but it would clearly not be appropriate with a context such as *this is not the time to be negative*. Likewise, extracting a phrase that appears in contexts very similar to that of an original phrase is limited by the effectiveness of the modeling of context used: for instance, several occurrences of *Spain defeated France* and *Spain lost to France* should not be used as evidence for establishing a paraphrasing relationship between the two verbs.[1]

In contrast, whenever parallel monolingual corpora aligned at the sentence level are available, the task of subsentential paraphrase acquisition can be cast as one of word alignment between two aligned sentences.[2] Previous works have exploited multiple translations, which occur very infrequently naturally. But such translations are sometimes produced, albeit in small quantities, for example as multiple reference translations for evaluating Machine Translation outputs automatically. Barzilay and McKeown [3] applied the distributionality hypothesis on such parallel sentences, and Pang *et al.* [16] proposed an algorithm to align sentences by recursive fusion of their common syntactic constituants. Callison-Burch *et al.* [6] describe an automatic metric that can be used to compare techniques extracting subsentential paraphrases from pairs of sentential paraphrases.

3 Acquisition of Subsentential Paraphrases from Sentential Paraphrases

As discussed in section 2, acquiring subsentential paraphrases is a challenging task. In this work, we consider the most simple scenario where sentential paraphrases are available and words/phrases from one sentence can be aligned to words/phrases from the other sentence. In this section we describe several techniques and their implementation that perform the task of subsentential unit alignment and how their results can be combined in a simple way. We will also describe an existing evaluation metric that will allow us to compare the performance of these techniques.

[1] As previously said, the use of *comparable* corpora provides a promising way to alleviate this issue, as limiting the corpus to, for example, press coverage for the same piece of news strongly increases the probability of finding very close contents.

[2] We do not address here the discourse and implication issues which make aligning the full contents of two such sentences not possible in all cases.

3.1 Statistical Word Alignment Method (Word)

In phrase-based Statistical Machine Translation, bilingual phrases are extracted from parallel corpora as the basic units for translating. These *biphrases* are often extracted in two steps [15] : first, word alignments are found in both directions, and some heuristics is then applied to symmetrize these alignments and to extract biphrases from the resulting alignment. Word alignment models are learnt from the full training set of parallel sentences in the training corpus, more data typically resulting in improved performance. Furthermore, because the underlying training algorithms make the assumption that sentences in a pair form a strict correspondance, sentences that are complete and somehow literal translations will make word alignment and biphrase extraction more precise.[3] Note that alignment models typically support alignment to NULL tokens for words which could not be aligned, thus representing word insertions/deletions. However, these words can still be included in the extracted biphrases depending on the heuristics used.

Using such alignment models for the monolingual case has already been tested, but to our knowledge no work has reported using parallel monolingual corpora due to their lack of availabibility.[4] For this work, we collected sentential paraphrases to build a monolingual parallel corpus. In order to increase the number of examples at the sentential level, we take all pairings for paraphrases belonging to the same group if they exist to build our training corpus.[5] We used the MOSES system [12] for word alignments (using GIZA++ [15]) and default symmetrisation. Once an alignment matrix is available, we extract biphrases corresponding to possible subsentential paraphrases using the following criterion [6]: two phrases from each sentence constitute a paraphrase pair if all words from one phrase are aligned to at least one word from the other phrase and not to words outside it.

3.2 Term Variant Identification Method (Term)

Sentential paraphrases can use common words, but also synonyms and more generally phrasal paraphrases. For such pairings, and under certain conditions that are met with the types of corpus that we use, rules can be expressed to model acceptable variations. Research on term variant extraction thus offers a direct solution to the problem of subsentential alignment for some units. We have

[3] This partly explains why some language pairs are harder to align than others.

[4] Works based on translational equivalence such as [1] alleviate this issue by using more readily available bilingual parallel corpora. However, one of the main limitation of this approach, which motivated works on context modeling for validating the extracted paraphrases [5,14], is that the extracted biphrases are only indirectly aligned. The strong limitation of our work for using parallel monolingual corpora finds here one of its main justifications.

[5] Note that this will give a clear advantage to the statistical word alignment technique over the other techniques that we will discuss, which do not currently support exploiting information from other sentences or other sentence pairings.

used the FASTR system [8], which takes as input a corpus and a list of terms and outputs the indexed corpus in which terms and variants are recognized, using sets of meta-rules applying to term rules to define acceptable variations. Meta-rules allow us to define morphosyntactic rewriting patterns as well as morphologic and semantic lexical relationships. FASTR offers a large ruleset (mostly for nominal and verbal terms) and large lists of morphological variants and synonyms.

The controlled indexing program of FASTR extracts all variants from a list of terms from a corpus. We used it to find phrase alignments in both directions. Given a pair of sentential paraphrases, all phrases up to a given length were extracted from one of the paraphrases. Those were then used by FASTR as input terms to perform controlled indexing of the corpus consisting solely of the other paraphrase. Only biphrases that are found in of both directions are kept. We modified the program configuration so that it accepts one-word terms, useful for synonym detection, but otherwise used its default resources. This consequently performs a biphrase extraction focussed on nominal and verbal terms.

3.3 Alignment by Syntactic Fusion Method (Synt)

The exploitation of the parallelism of two sentential paraphrases can be pushed further: if two such paraphrases share their high-level syntactic structure, then it is possible to guide the alignment of their words by recursively aligning their syntactic subconstituents. Pang *et al.* [16] proposed an implementation of this idea, which is illustrated on Figure 1.

The two sentences of a paraphrase pair are first syntactically parsed. Fusion then takes place in the following way: two syntactic subtrees are recursively merged if their root category and the categories of the ordered list of their daughter subtrees match. Otherwise, when the categories of the daughter subtrees do not match, a list of alternative derivations is created at that node.

In order to avoid mistakenly merging some subtree configurations, the authors introduced a *lexical blocking* mechanism, which prevents merging two subtrees if a content that belongs to the accessible vocabulary of one daughter subtree of the first subtree also belongs to the accessible vocabulary of a different daughter subtree of the second subtree. This prevents merging in cases such as active and passive voice sentential paraphrases, where in spite of matching high-level syntactic structures the agent and patient have been swapped.

In a last step, the obtained parse forest is linearized to yield a word lattice. This lattice can finally be reduced by merging edges with the same words that have common prefixes or suffixes.

All subpaths originating from the same start and end nodes thus represent subsentential units in an equivalence relation.[6] The set of all pairs of subsentential paraphrases encoded in the lattice can be effectively extracted by a simple traversal of the lattice. Note that the presented algorithm can work with any number of input sentential paraphrases as illustrated in the original article [16].

[6] Note that in the extreme case of two sentences whose root nodes cannot be merged the smallest pair of equivalent units is made up the two full sentences.

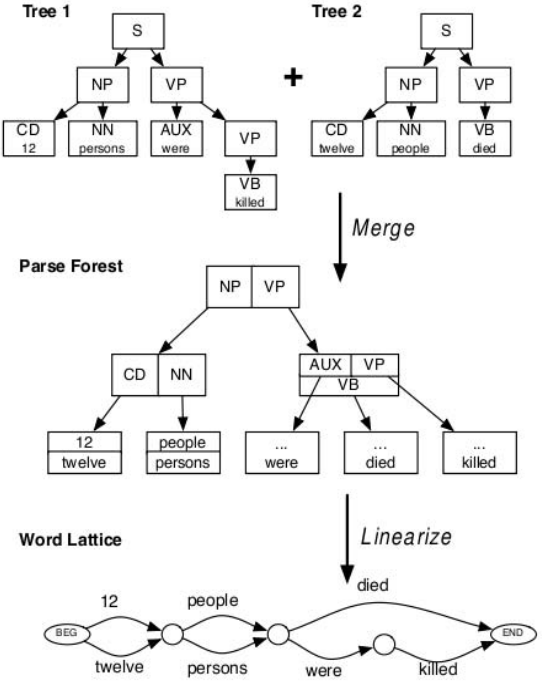


Fig. 1. Illustration of Pang *et al.*'s [16] syntactic fusion algorithm

Our implementation of this technique revealed several limitations. First, this algorithm stops all merging whenever lexical blocking is fired, which was probably motivated by the fact that the objective of the authors was to generate new sentential paraphrases from a set of existing sentences. However, because we want to extract as many (accurate) pairings as possible, we allow merging to continue on subconstituents which should not be affected by lexical blocking.

We also addressed the strong dependency of the algorithm on the precision of the automatic syntactic parses used. Indeed, we observed many cases in our development data where legitimate merging did not take place because of incompatible parse trees resulting from locally wrong parse trees. However, wrong syntactic parses can sometimes produce good alignments if both parses can be merged in the same way as valid parses would allow. Using the Berkeley probabilistic parser [10], the k-best parses for both sentences are used, and the i^{th} parse of the first sentence and the j^{th} parse of the second one kept are those for which the correponding lattice is the most compact before reduction.

The intuitive motivation for this choice is that the more compact a lattice is before reduction the more two sentences have been aligned (which is a sought property given the parallelism of the input sentences), and that the reduction operation should reduce as few as possible nodes that would not have been previously merged due to compatible subparses of the two sentences. We chose

a value of 5 for k, thus limiting the number of merged configurations to $5^2 = 25$ per sentence pair.

3.4 Combination of Paraphrase Pairs from Different Techniques

All the presented techniques for extracting sub-sentential alignments were run independently to produce candidates for phrases up to 7 tokens (pairs of identical phrases excluded). Each technique makes use of its own hypotheses and resources, and it can be expected that they could have different performances depending on the configurations of the merged sentences. Working on efficiently combining these techniques is therefore an interesting issue. In this work, we started by considering the simple case of output combination by performing set unions and intersections. Taking the union of the candidate pairs of two or more techniques is expected to increase the *recall* of found pairs relative to a reference alignment, while taking the intersection is expected to increase the *precision* of such pairs. It is of particular interest to measure how a combination of both measures would behave for such cases.

4 Experimental Framework

4.1 Corpus Collection

In order to build development and test corpora, we set up a web-based data acquisition experiment. Volunteer contributors were asked to translate sets of sentences into French[7]. The same sets of sentences were translated by several contributors, from any of 10 languages from the Europarl corpus [11] of European parliamentary debate. The web interface provided the contributors, who were not for the most part professional translators, with several convenience tools to help them in their task. One of them allowed checking a reference translation from Europarl, which was not used to build our corpus, to make local improvements and corrections once an initial candidate translation was submitted. This technique proved quite successful in ensuring an acceptable quality for most submitted translations. All translations used in these experiments were manually checked by a unique annotator who followed the rule to remove any translation which showed too strong a bias towards the reference translation between the two versions of a translation.

In order to measure the similarity degree between lexical paraphrases obtained from different languages, we computed the *overlap coefficient*, which represents the lexical overlap percentage between the vocabularies of two sentences S_1 and S_2:

$$CO = \frac{|S_1 \cap S_2|}{min(|S_1|, |S_2|)} \quad (1)$$

[7] We considered important to use for our experiments a language for which all our contributors and human annotators had a native or near-native command.

Table 1. Similarity degree between paraphrases obtained from different languages

	All tokens					Lemmas of content words				
	en	**es**	**de**	**it**	**pt**	**en**	**es**	**de**	**it**	**pt**
en	0.90 $_{172}$	0.64 $_{69}$	0.59 $_{89}$	0.63 $_{84}$	0.62 $_{58}$	0,90 $_{172}$	0,65 $_{69}$	0,61 $_{89}$	0,66 $_{84}$	0,64 $_{58}$
es	*	-	0.62 $_{57}$	0.63 $_{57}$	0.64 $_{51}$	*	-	0,57 $_{57}$	0,68 $_{57}$	0,68 $_{51}$
de	*	*	-	0.58 $_{67}$	0.61 $_{53}$	*	*	-	0,59 $_{67}$	0,62 $_{53}$
it	*	*	*	-	0.65 $_{50}$	*	*	*	-	0,66 $_{50}$
pt	*	*	*	*	-	*	*	*	*	-

The minimum number of pairs for a group was set to 20 in this experiment. The left side of Table 1 shows the average coefficient of lexical overlap for all tokens on the selected groups of paraphrases and for different source languages. Numbers shown as indices represent the number of sentential paraphrases (common translations) obtained from two languages. For instance, the 172 paraphrases obtained from English have an average of 90% common tokens. In contrast, we find that those from two different languages contain on average between 36% and 42% different tokens. These values show, as we expected, that we obtain more lexical variation using different source languages for semantically equivalent sentences.

We repeated this experiment considering this time only lemmatised forms of content words. Results, shown on the right side of Table 1, show a similarity degree which is slightly higher than in the previous experiment, varying from 57% to 68% for different languages. This still shows a significant level of lexical variation in the translation process at the level of the content words used.

In order to simulate various degrees of parallelism between two sentences in a pair, we built two sub-corpora from our full corpus: we took a set of 50 sentences which were selected on the basis that 4 independent valid translations from English and one valid translation from German, Spanish, Italian and Portuguese were available.[8] We therefore obtained two corpora of 50 groups of 4 paraphrases each. In each group, one paraphrase is randomly selected as a "reference paraphrase" to which the three others will be aligned. Three human annotators were then asked to manually align at the token level each of the 300 sentence pairs (2 corpora x 50 groups x 3 alignments). The YAWAT [7] manual word alignment tool was used, to allow aligning sentences by visual selection of phrases and optional checking on word alignment matrices. Each sentence pair was annotated by a single annotator, as the work of Callison-Burch *and al.* [6] reports an acceptable inter-annotator agreement rate on such a task.[9] We nevertheless asked one of the judges to check all alignments and make the necessary modifications to make them more uniform. Reference biphrases were finally automatically extracted from the token alignment matrix by following the rule previously described at the end of section 3.1.

[8] Note that the version of the Europarl corpus that we used did not contain information about the original language for sentences.

[9] It should be noted that their work was on news text in English and that their annotators had been provided with a detailed annotation guide.

Source	Contributed translation
English	Plusieurs orateurs ont considéré que ceci est trop tardé.
English	Plusieurs locuteurs ont jugé cela nécéssaire depuis longtemps.
English	Plusieurs orateurs ont consideré que cela aurait dû être fait depuis longtemps.
English	Beaucoup d'orateurs considèrent qu'il s'agit d'un processus qui aurait du être conclu il y a longtemps.
German	Plusieurs orateurs les croient obligatoires depuis si longtemps.
Spanish	Plusieurs intervenants l'ont considéré comme une chose indispensable.
Italian	Le retard avec lequel s'accomplie cette étape a été souligné dans de nombreuses interventions.
Portuguese	Beaucoup d'orateurs considèrent qu'il s'agit d'un processus qui aurait du être conclu il y a longtemps.

Fig. 2. Example of a multiply-translated sentence from English and from German, Spanish, Italian and Portuguese

An example of a paraphrase group is shown on Figure 2. As can be seen, the source language often implies an important bias for the production of the contributed translation, which results in part from the fact that our contributors were not professional translators. One can also notice that some original translations may express slightly different content resulting from the choices of the sequence of translators involved to obtain these translations.

4.2 Experimental Results

In order to evaluate and compare the results of the implemented techniques for subsentential paraphrase extraction, we followed the PARAMETRIC approach described in [6]: the set of candidate biphrases extracted from a sentence pair is compared with a set of reference biphrases obtained through human annotation (as described in section 4.1) by computing *precision* and *recall* values. The former value corresponds to the proportion of candidates produced by a technique which are correct relative to the reference biphrases, while the latter value corresponds to the proportion of reference biphrases that are extracted by a technique. As we are also interested in the combination of both these values when combining candidate sets, we also computed an F-measure value, F_1, which considers recall and value as equally important. We run the three techniques described in section 3 on our two subcorpora, denoted **en2fr** and **xx2fr**, and computed evaluation scores on their result sets and on result sets obtained for simple combinations. Results are given on Table 2.

It is first quite apparent that the performance of all techniques, both in terms of precision and recall, is highly dependent on the nature of the sentential paraphrase pairs, which can be interpreted as a higher complexity for aligning sentence pairs produced from different languages. If SYNT is unsurprisingly very sensitive to this, WORD also seem to be significantly impacted when attempting alignment on less literal sentences, which can be compared to the higher difficulty in aligning unrelated languages during training in Statistical Machine Translation.

Table 2. Results obtained for each technique and some combinations of their outputs in terms of precision (**P**) and recall (**R**) of their candidate biphrases relative to reference biphrases. The $\mathbf{F}_1$ values gives as much importance to precision and recall.

	WORD	TERM	SYNT	T∪W	T∩W	T∪S	T∩S	W∪S	W∩S	T∪W∪S
Paraphrases obtained by translating from English (**en2fr**)										
P	41.94	41.19	50.16	41.54	55.97	46.48	80.76	40.83	71.21	40.46
$\mathbf{R}_{/3578}$	67.07	3.07	8.77	67.66	2.48	11.26	0.58	67.83	8.02	68.41
$\mathbf{F}_1$	51.61	5.87	14.93	51.47	4.76	18.13	1.16	50.98	14.41	50.58
Paraphrases obtained by translating from 4 languages (**xx2fr**)										
P	27.05	35.98	40.46	27.08	42.98	39.46	28.57	26.91	50.15	26.90
$\mathbf{R}_{/2517}$	51.80	3.05	8.26	52.92	1.94	11.08	0.23	53.43	6.63	54.39
$\mathbf{F}_1$	35.54	6.07	13.72	35.83	3.72	17.30	0.47	35.79	11.72	36.00

Looking at recall, we can note a strong difference between WORD on the one side, and TERM and SYNT on the other side, with the latter two proposing much fewer paraphrase pairs from the reference alignments. The proportion of aligned words is unsurprisingly higher for WORD, as this statistical word alignment technique attempts aligning as many words as possible, although aligning to a NULL token is possible under certain circumstances. Nonetheless, WORD still achieves a reasonable precision score. Note, however, that this technique benefited from a positive bias as it was able to exploit all sentential paraphrase pairings to build its alignment model, and therefore could effectively make use of redundancy while the two other techniques could not take such information into account as implemented. TERM seems to be specialized in extracting very focused biphrases. SYNT achieves the best precision overall, with a 10-point advantage for the paraphrases obtained from one language over paraphrases obtained from 4 different languages.

Para 1 (**German**)	En ce qui concerne les relations internationales , la communauté doit s' y attaquer de manière déterminée et s' accorder avec la politique extérieure
Para 2 (**Italian**)	Quant aux relations internationales , la Communauté est confrontée aux décisions relatives à la politique étrangère
REF	**(En ce qui concerne↔Quant aux) (En ce qui concerne les relations↔Quant aux relations) (la politique extérieure↔la politique étrangère) (politique extérieure↔politique étrangère) (extérieure↔étrangère)**
WORD	(En↔Quant) (ce↔à) (concerne les↔aux) (concerne les relations↔aux relations) (concerne les relations internationales↔aux relations internationales) (concerne les relations internationales ,↔aux relations internationales ,) (concerne les relations internationales , la↔aux relations internationales , la) (la communauté doit s' y attaquer↔la Communauté est confrontée aux décisions) (communauté doit s' y attaquer↔Communauté est confrontée aux décisions) (doit s' y attaquer↔est confrontée aux décisions) (déterminée↔relatives) **(la politique extérieure↔la politique étrangère) (extérieure↔étrangère)**
TERM	**(politique extérieure↔politique étrangère) (extérieure↔étrangère)**
SYNT	**(En ce qui concerne les↔Quant aux)**

Fig. 3. Examples of *bi-phrases* extracted by different techniques from a pair of paraphrases produced from German and Italian sentences (biphrases in bold belong to the reference set)

The various tested combinations reveal expected gains in recall for union and in precision for intersection. Accordingly, on the **en2fr** corpus, maximum precision is obtained by computing intersection sets with the results of SYNT, and maximum recall is obtained by computing union sets with the results of WORD. Results are roughly similar on the **xx2fr** corpus, with the notable exception of the T∩S combination which obtained a comparatively much worse performance than on the other corpus.

Figure 3 illustrates an example of alignment results between two paraphrases obtained from German and Italian, whose alignment is difficult as confirmed by the low number of biphrases in the reference set. WORD was unable to reliably align the words and produced many incorrect biphrases. TERM and SYNT have instead proposed only few candidates, which reflects again the difficulties of matching encountered by these two techniques.

5 Conclusion and Future Work

In this article, we have described the task of subsentential paraphrase extraction from sentential paraphrases, a resource which is rare but which allows us, as we argued, to concentrate on the most natural scenario for observing such local paraphrases. Furthermore, such sentential paraphrases allow us to trivially extract contextual information (e.g. words linked by a grammatical dependency to words in the paraphrase pair) associated to paraphrase pairs that can be used to bootstrap context profiles for which the paraphrase pair is valid.

We have described three techniques, initially developed for different purposes, which operate at various levels and use different resources, and compared them on two subcorpora representing two levels of parallelism for sentences. Acceptable levels of precision and recall relative to a reference alignment were achieved, and simple combinations of results yielded gains for one of the two metrics.

Our future work will be organized along three different lines. First, we want to be able to generalize the obtained subsentential paraphrases to learn paraphrasing patterns which integrate contextual information. Then, we plan to first independently improve each of the presented techniques and then work on efficient hybrid implementations at extraction time. Finally, we want to study techniques for validating paraphrases acquired on monolingual parallel corpora on much more readily available monolingual comparable corpora.

Acknowledgements. This work was supported by a grant from LIMSI. The authors wish to thank the volunteer contributors who took part in the data collection.

References

1. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proceedings of ACL, Ann Arbor, USA (2005)
2. Barzilay, R., Lee, L.: Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In: Proceedings of NAACL-HLT, Edmonton, Canada (2003)

3. Barzilay, R., McKeown, K.: Extracting paraphrases from a parallel corpus. In: Proceedings of ACL, Toulouse, France (2001)
4. Bhagat, R., Ravichandran, D.: Large scale acquisition of paraphrases for learning surface patterns. In: Proceedings of ACL-HLT, Columbus, USA (2008)
5. Callison-Burch, C.: Syntactic constraints on paraphrases extracted from parallel corpora. In: Proceedings of EMNLP, Hawai, USA (2008)
6. Callison-Burch, C., Cohn, T., Lapata, M.: Parametric: An automatic evaluation metric for paraphrasing. In: Proceedings of COLING, Manchester, UK (2008)
7. Germann, U.: Yawat:Yet Another Word Alignment Tool. In: Proceedings of the ACL 2008: HLT Demo Session, Columbus, Ohio, pp. 20–23 (2008)
8. Jacquemin, C.: Syntagmatic and paradigmatic representations of term variation. In: Proceedings of ACL, College Park, USA (1999)
9. Kauchak, D., Barzilay, R.: Paraphrasing for automatic evaluation. In: Proceedings of NAACL-HLT, New York, USA (2006)
10. Klein, D., Manning, C.: A* parsing: Fast exact viterbi parse selection. In: Proceedings of NAACL-HLT, Edmonton, Canada (2003)
11. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT summit, Citeseer, vol. 5 (2005)
12. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of ACL, demo session, Prague, Czech Republic (2007)
13. Lin, D., Pantel, P.: Discovery of inference rules for question answering. Natural Language Engineering 7(4), 343–360 (2001)
14. Max, A.: Local rephrasing suggestions for supporting the work of writers. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 324–335. Springer, Heidelberg (2008)
15. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics (2003)
16. Pang, B., Knight, K., Marcu, D.: Syntax-based alignement of multiple translations: Extracting paraphrases and generating new sentences. In: Proceedings of NAACL-HLT, Edmonton, Canada (2003)
17. Zhao, S., Wang, H., Liu, T., Li, S.: Pivot approach for extracting paraphrase patterns from bilingual corpora. In: Proceedings of ACL-HLT, Columbus, USA (2008)

Digital Learning for Summarizing Arabic Documents

Mohamed Mahdi Boudabous[1], Mohamed Hédi Maaloul[2], and Lamia Hadrich Belguith[1]

[1] ANLP Research Group, MIRACL Laboratory, Faculty of Economic Sciences and Management of Sfax (FSEGS) - B.P.1088, 3018 Sfax, Tunisia
[2] LPL Laboratory, CNRS-Université de Provence - 5 Avenue Pasteur 13604 Aix en Provence – France
mahdiboudabous@gmail.com, mohamed.maaloul@lpl-aix.fr, l.belguith@fsegs.rnu.tn

Abstract. We present in this paper an automatic summarization method of Arabic documents. This method is based on a numerical approach which uses a semi-supervised learning technique. The proposed method consists of two phases. The first one is the learning phase and the second is the use phase. The learning phase is based on the Support Vector Machine (SVM) algorithm. In order to evaluate our method, we conducted a comparative study that involves the results generated by our system AIS (Arabic Intelligent Summarizer) with that realized by a human expert. The obtained results are very encouraging and we plan to extend our evaluation on a larger corpus to ensure the performance of our system.

Keywords: Automatic summarization, Arabic documents, Machine Learning, Numerical approaches.

1 Introduction

In the current context, we have to deal with a huge mass of electronic textual documents available through the net. We need tools offering fast visualization of the documents (so that the user can evaluate its relevance). Automatic summarization provides a solution which makes it possible to extract interesting information for an advantageous reuse. Indeed, the summary helps the reader to decide whether the original document contains the required information or not. Moreover, in some cases the reader does not need to read the totality of the original document, simply because the required information is in the summary [1].

Automatic summarization approaches are inspired by various orientations. Some approaches rely on symbolic techniques (based on the analysis of the discourse and its discursive structure), some others are based on numerical treatments (based on statistical, or even on learning) [2].

In addition, the majority of automatic summarization systems mainly treat documents in Indo-European languages such as English and French. To our knowledge, there are only few implementations of these methods on Arabic language, such as LAKHAS [3] and Al Lakas El'eli [4]. Thus, there is an increasing need to develop

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 79–84, 2010.

automatic summarization systems dedicated to Arabic to handle the increasing amount of electronic documents written in Arabic [1].

Thus, the achievements in the field of automatic summarization are generally set out again according to the used approaches. Mainly three approaches are distinguished: numerical, symbolic and hybrid. Our contribution is in the context of numerical approach and we propose a system for the automatic summarization of Arabic documents which is based on a purely Machine learning (ML) technique: ML technique within the framework classification, is shown to be a promising way to combine automatically sentence features [5]. In our method, a classifier is trained to distinguish between two classes of sentences: summary and non-summary ones.

Statistical features that we consider in this work are partly from the state-of-art, and they include cue-sentencess and positional indicators [6], title-keyword similarity [7], and other features.

This paper is structured around four sections: Section 1 presents most related works to ours. Section 2 exposes the proposed method and the summarizing workflow and Section 3 describes the implementation of our approach and the primary results. Section 4 presents the conclusion and the future works.

2 Related Work

Three approaches have been proposed to the summarizing of documents: Linguistic approaches based on a formal representation of knowledge contained in documents or on a reformulation technique. Indeed, these approaches are usually a formal representation of knowledge contained in documents or on reformulation techniques. Numerical approaches are based on calculating a score associated for each sentence to estimate its importance relative to other sentences of the document. This score is calculated by using various statistical methods, probabilistic and learning. Hybrid approaches combine the previous approaches to improve the quality of the summary.

In this paper, we explore a numerical approach and present some examples. Numerical approaches are essentially based on calculating a score associated for each sentence to estimate its importance. The final summary will only keep the sentences that have the highest scores.

There are two main techniques: statistical and learning techniques. Recently, various authors have explored Machine Learning techniques to summarize documents [7]. This is thanks to the best performance of these techniques.

The learning techniques are classified into three classes. The first class is the supervised learning, this class is based on two phases: the learning phase that use a training corpus of a very large size and the validation phase that use another corpus called validation corpus [8]. The second class is the semi-supervised learning that has only a learning phase; this phase requires a training corpus of small size [9] [10]. The third one is the non-supervised learning, which does not require either a training corpus or a validation corpus.

The numerical approaches can be applied to all types of corpus and can operate in a big number. The most important systems which are based on the numerical approaches are: LAKHAS system [3] which summarizes Arabic documents in XML format. CBSEAS “Clustering-Based Sentence Extractor for Automatic Summarization” system

[11] treats the case of multi-document summary. Its principle is that the more redundant information are the more important they will be.

Our method treats the numerical approaches that have proven their effectiveness in other languages. More precisely, we use Machine learning techniques based on semi-supervised learning; this choice is justified by the fact that it allows involving a system with only a small number of labeled sentences and a large number of not labeled ones.

3 Proposed Method

In this section, we present an overview of the proposed method and the summarizing workflows for the HTML documents.

3.1 An Overview of Our Method

We propose a new method for the automatic summarization of the newspaper articles in Arabic language. It is based on a Machine learning technique. More precisely it is based on the semi-supervised learning technique which is composed of two phases: the first one is the learning phase which allows the system to learn how to extract summary sentences. We use Support Vector Machines algorithm (SVM) for this phase. The second phase is the use phase which allows users to summarize a new document. Fig. 1 presents the details of the proposed method and the two phases.

3.2 Summarization Workflow

3.2.1 The Learning Phase

In this phase, the system designer should provide the training corpus and the extraction features to perform the learning.

The training corpus is composed of the source documents and their summaries. All the documents are initially pretreated to prepare their segmentation in titles, sections, paragraphs and sentences. This segmentation is based on the criteria of punctuation and HTML tags. After the segmentation step, each sentence of the segmented document will be notified according to some features. This step leads to the construction of a set of the vectors V corresponding to the values of the specific features to the sentence. These vectors are called extraction vectors or score vectors. Each vector is associated with a Boolean criterion which indicates the sentence class: summary or non-summary.

The extraction vector has the following structure: V1 (*S1, S2, S3… Sn*), where *Si* is the score of the criterion *i* and *n* is the number of the criteria.

In the learning phase, extraction vectors are combined to associate a score with each feature and generate rules.

3.2.2 The Use Phase

In this phase, the user provides a HTML document as an input for the system. This document is segmented and notified in order to generate a set of extraction vectors. The system uses the generated rules to classify each sentence.

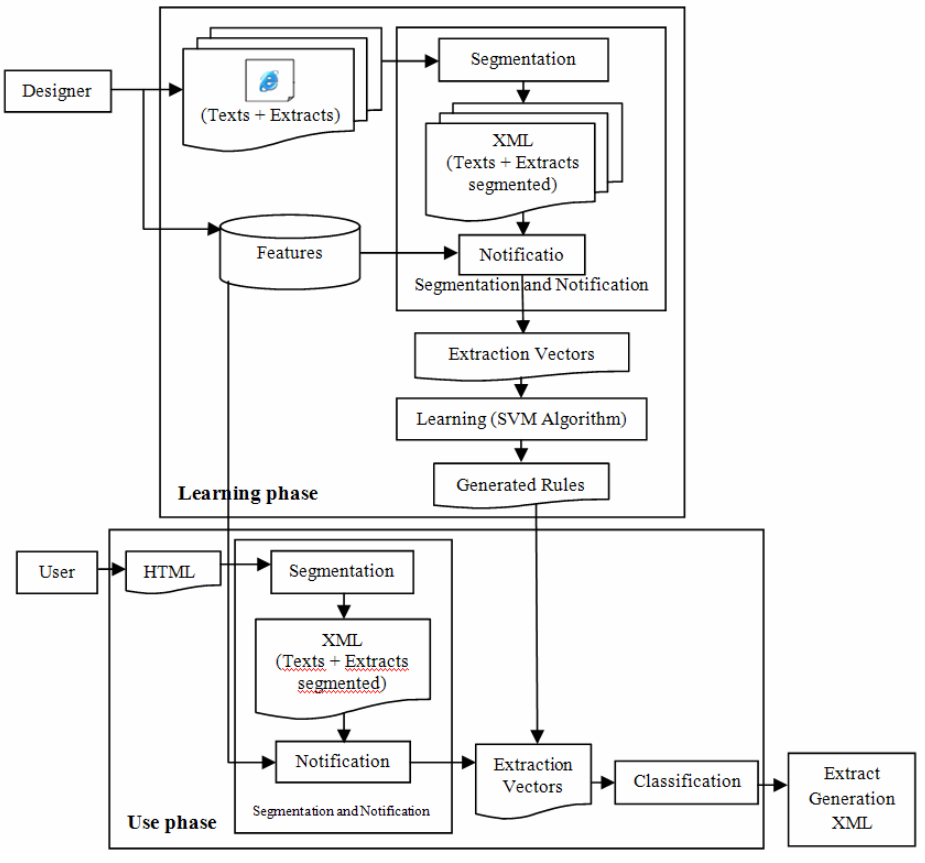


Fig. 1. The principle of the proposed method

4 The AIS System

The method that we proposed for automatic summarization of Arabic documents has been implemented through the AIS (Arabic Intelligent Summarizer) system. In this section, we present the implementation details and the preliminary results.

4.1 Implementation Details

Our corpus is composed of 500 Arabic documents collected from the web. These documents represent newspaper articles selected according to various orientations (sport, economy, education, etc.). The newspaper articles are of HTML type with a UTF-8 coding. The summaries of these documents are produced by three human experts. Then, we use the index of kappa[1] to calculate the similarity between human experts and generate one summary for each document.

[1] http://kappa.chez-alice.fr/

After the segmentation step, we use 15 features to classify each sentence. Some of these features are detailed in Table 1.

Table 1. Features details

Features	Details
Position in the text	Indicates the position of the sentence in the text.
First sentence in the section	Indicates if the sentence is the first in the section or not.
First sentence in the paragraph	Indicates if the sentence is the first in the paragraph or not.
Range of the paragraph	Indicates the range of paragraph that contains the sentence.
Tf_idf score	Calculates the tf*idf of the score.
Tf score	Calculates the Tf of the score.
Title keywords	Presents the number of title keywords in the sentence.
Indicative expressions	Presents the number of indicative expressions in the sentence.

Finally, we obtain a file that contains the set of extraction vectors which constitute the input of the learning phase. In the learning phase, we use the SVM algorithm to learn how to classify the summary and non-summary sentences. At the end of the learning phase, a score is associated with each feature. Some features can have a score of zero. The SVM algorithm generates a rule by summing scores associated with each feature.

The system uses the generated rules to calculate the score of each sentence. If the score is positive, the sentence will be considered as a summary sentence, otherwise the sentence is considered as a non-summary sentence. Finally, the system combines summary sentences to obtain the summary.

4.2 Preliminary Results

We used 60 documents of our corpus to experiment our system (i.e. 50 documents for the learning phase and 10 documents for the evaluation phase). The obtained summaries are compared to the human summaries. The average measures for Precision, Recall and F-measure are respectively 0.992, 0.991and 0.991 (see Table 2).

Table 2. Evaluation results

	Precision	Recall	F-measure
Weighted Avg	0.992	0.991	0.991

5 Conclusion and Future Work

In this paper, we have proposed a method for automatic summarization of Arabic documents. Our method is implemented by AIS system and is based on the Machine learning technique. Our work focuses on a particular type of documents (i.e., the newspaper articles in HTML format). We believe that the preliminary results are very encouraging. Indeed the F-measure is equal to 0.991. We note that we used a small

corpus for the evaluation but as perspectives, we plan to extend the evaluation on a larger corpus.

We also intend to apply the proposed method for other types of documents such as XML and TXT.

References

1. MaÃloul, M.H., Ellouze Khemakhem, M., Belguith Hadrich, L.: Al Lakas El'eli /ÇááÎÇÕ ÇáÂáí: Un système de résumé automatique de documents arabes. International Business Information Management Association (IBIMA 2008) (2008)
2. Amini, M.R., Gallinari, P.: Apprentissage numérique pour le résumé de texte. Les Journées d'Étude de l'ATALA, Le résumé de texte automatique: solutions et perspectives (2003)
3. Douzidia, S., Lapalme, G.: Lakhas, an Arabic summarization system. In: Proceedings of the HLT-NAACL Workshop on Text Summarization DUC 2004 (2004)
4. Maâloul, M.H., Ellouze Khemakhem, M., Belguith Hadrich, L.: Proposition d'une méthode de résumé automatique de documents arabes. GEI 2006 (2006)
5. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the 18th ACM SIGIR Conference (1995)
6. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research and Development (1958)
7. Alrahabi, M., Mourad, G., Djioua, B.: Filtrage sémantique de textes en arabe en vue d'un prototype de résumé automatique. In: JEP/TALN 2004 (2004)
8. Mani, I., Bloedorn, E.: Machine Learning of Generic and User-Focused Summarization. In: Proceedings of the Fifteenth National Conference of Artificial Intelligence, AAAI 1998 (1998)
9. Amini, M.R.: Apprentissage automatique et recherche de l'information: application à l'extraction d'information de surface et au résumé de texte. Thèse de doctorat (2001)
10. Amini, M.R., Gallinari, P.: The Use of Unlabeled Data to Improve Supervised Learning for Text Summarization. In: SIGIR (2002)
11. Bossard, A., Généreux, M., Poibeau, T.: CBSEAS, a Summarization System Integration of Opinion Mining Techniques to Summarize Blogs (2009)

Concept Based Representations for Ranking in Geographic Information Retrieval*

Maya Carrillo[1,2], Esaú Villatoro-Tello[1], Aurelio López-López[1], Chris Eliasmith[3], Luis Villaseñor-Pineda[1], and Manuel Montes-y-Gómez[1]

[1] Coordinación de Ciencias Computacionales, INAOE
Luis Enrique Erro 1, Santa Maria Tonantzintla, Puebla, México, C.P.72840
{cmaya,villatoroe,allopez,villasen,mmontesg}@inaoep.mx
[2] Facultad de Ciencias de la Computación, BUAP
Av. San Claudio y 14 Sur Ciudad Universitaria, 72570 Puebla, México
[3] Centre for Theoretical Neuroscience, University of Waterloo
200 University Ave., Waterloo, Ontario, Canada
celiasmith@uwaterloo.ca

Abstract. Geographic Information Retrieval (GIR) is a specialized Information Retrieval (IR) branch that deals with information related to geographical locations. Traditional IR engines are perfectly able to retrieve the majority of the relevant documents for most geographical queries, but they have severe difficulties generating a pertinent ranking of the retrieved results, which leads to poor performance. A key reason for this ranking problem has been a lack of information. Therefore, previous GIR research has tried to fill this gap using robust geographical resources (i.e. a geographical ontology), while other research with the same aim has used relevant feedback techniques instead. This paper explores the use of Bag of Concepts (BoC; a representation where documents are considered as the union of the meanings of its terms) and Holographic Reduced Representation (HRR; a novel representation for textual structure) as re-ranking mechanisms for GIR. Our results reveal an improvement in mean average precision (MAP) when compared to the traditional vector space model, even if Pseudo Relevance Feedback is employed.

Keywords: Geographic Information Retrieval, Vector Model, Random Indexing, Context Vectors, Holographic Reduced Representation.

1 Introduction

Geographic Information Retrieval (GIR) deals with information related to geographic locations, such as the names of rivers, cities, lakes or countries [18].

* The first and second authors were supported by Conacyt scholarships 208265 and 165545 respectively, while the third, fifth and sixth authors were partially supported by SNI, Mexico. This work has been also supported by Conacyt Project Grant 61335.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 85–96, 2010.

Information that is related to a geographic space is called geo-referenced information, which is often linked to locations expressed as place names or phrases that suggest a geographic location. For instance, consider the query: "ETA in France". Traditional IR techniques will not be able to produce an effective response to this query, since the user information need is very general. Therefore, GIR systems have to interpret implicit information contained in documents and queries to provide an appropriate response to a query. This additional information would be needed in the example to match the word "France" with other French cities as Paris, Marseille, Lyon, etc.

Recent developments in GIR systems have demonstrated that the GIR problem is partially solved through traditional or minor variations of common IR techniques. It is possible to observe that traditional IR engines are able to retrieve the majority of relevant documents for most geographical queries, but they have severe difficulties generating a pertinent ranking of the retrieved results, which leads to poor performance.

An important source of the ranking problem has been the lack of information. Therefore, previous research in GIR has tried to fill this gap using robust geographical resources (i.e. a geographical ontology), whilst other research has used relevance feedback techniques instead.

As an alternative, our method suggests representing additional information incorporating concept-based representations. We think that concept-based schemes provide important information, and that they can be used as a complement to the Bag of Words representations. Our goal is therefore to investigate whether combining word-based and concept-based representations can be used to improve GIR.

In particular, we consider the use of two document representations: a)Bag of Concepts (BoC), as proposed by Sahlgren and Cöster [3], to represent a document as the union of the meanings of its terms; b)Holographic Reduced Representation (HRR) defined by Plate [2] to include syntactic structure. The purpose is to represent relations that give different ideas of location like: *in Paris, near Paris, across Paris*. This representation can help to state specific information for GIR.

The proposed BoC and HRR representations are vector representations constructed through the aid of Random Indexing (RI), a vector space methodology proposed by Kanerva et al [20].

The remainder of this paper is organized as follows. In Section 2 we briefly review some GIR related work. Section 3 presents Random Indexing word space technique. Section 4 describes the Bag of Concepts representation. Section 5 introduces the concept of Holographic Reduced Representations (HRRs) and presents how to use them to represent documents according to their spatial relations. Section 6 explains the experimental setup. Section 7 shows the results obtained with Geo-CLEF collections and queries from 2007 to 2008. Finally, Section 8 concludes the paper and gives some directions for further work.

2 GIR Related Work

Geographical Information Retrieval (GIR) considers the search for documents based not only on conceptual keywords, but also on spatial information (i.e.,

geographical references) [18]. Formally, a geographic query (geo-query) is defined by a tuple ⟨*what, relation, where*⟩ [19]. The *what* part represents generic terms (non-geographical terms) employed by the user to specify its information need, which is also known as the thematic part. The *where* term is used to specify the geographical areas of interest. Finally, the *relation* term specifies the "spatial relation", which connects *what* and *where*. For example in query: *Child labor in Asia*, the *what* part would be: *Child labor*, the *relation* term would be *in* and the *where* part *Asia*.

GIR was evaluated at the CLEF forum [14] from 2005 to 2008, under the name of the 'GeoCLEF' task [15]. Several approaches were focused on solving the ranking problem during these years. Common employed strategies are: a) query expansion through feedback relevance [6], [9], [10]; b) re-ranking retrieved elements through adapted similarity measures [7]; and c) re-ranking through information fusion techniques [9], [10], [11].

These strategies have been implemented following two main paths: first, techniques that have paid attention to constructing and including robust geographical resources in the process of retrieving and/or ranking documents. And second, techniques that ensure that geographical queries can be treated and answered by employing very little geographical knowledge.

As an example of those in the first category, previous research employed geographical resources in the process of query expansion. Here, they first recognize the geographical named entities (geo-terms) in the given geo-query by employing a GeoNER[1]-system. Afterwards, they then employ a geographical ontology to search for these geo-terms, and retrieve some other related geographical terms. The retrieved terms are then used as feedback elements to the GIR engine. However, a major drawback with these approaches is the huge amount of work needed in order to create such ontologies: for instance, Wang et al. in [6] employ two different geographical taxonomies (Geonames[2] and WorldGazetteer[3]) to construct a geographical ontology with only two spatial relations: *"part-of"* and *"equal"*. This leads to the fact that the amount of geographical information included in a general ontology is usually very small, which limits it as an effective geographical resource. Some other approaches that focus on the re-ranking problem propose algorithms that consider the existence of Geo-tags[4]; therefore, the ranking function measures levels of topological space proximity, or geographical closeness among the geo-tags of retrieved documents and geo-queries [7]. In order to achieve this, geographical resources are needed. Although these strategies work well for certain type of queries, in real world applications neither "geo-tags" nor robust geographical resources are always available.

In contrast, approaches that do not depend on any geographical resource, have proposed and applied variations of the query expansion process via relevance

[1] Geographical Named Entity Recognizer.
[2] Geonames geo coding web service: http://www.geonames.org/
[3] WorldGazetteer: http://www.world-gazetteer.com
[4] A Geo-tags is a label that indicates the geographical focus of certain document or geographical query.

feedback without special consideration for geographic elements [8], [9]. Despite this, they have achieved acceptable performance results, sometimes even better than those obtained employing resource-based strategies. There is also work focusing on the re-ranking problem; it considers the existence of several lists of retrieved documents from one or more IR engines. For instance, one IR engine can be configured to manage a thematic index (i.e., non geographical terms), while another IR engine is configured to manage only geographical indexes [8], [9], [10], [11], [18]. Therefore, the ranking problem is seen as an information fusion problem; where simple strategies only apply logical operators to the lists (e.g., AND) in order to generate one final re-ranked list [10], while others apply techniques based on information redundancy (e.g., CombMNZ, Round-Robin or Fuzzy Borda)[8], [10], [11], [18].

Recent evaluation results indicate that there is not a notable advantage of resource-based strategies over methods that do not depend on any geographical resource [11]. Motivated by these results, our method does not depend on the availability of geographical resources, but we contemplate the use of different lists of ranked retrieved documents (VSM, BoC and HRR) looking for improvement of the base ranker efficiency by the combination.

This work differs from previous efforts in that we consider, in the re-ranking process, the context information and syntactic structure contained in geo-queries and retrieved documents. This additional information is captured by BoC and HRR representations, which need special vectors, built by Random Indexing (RI).

3 Random Indexing

The vector space model (VSM) [16] is probably the most widely known IR model, mainly because of its conceptual simplicity and acceptable results. The model creates a space in which both documents and queries are represented by vectors. This vector space is represented by V a n x m matrix, known as term-document matrix, where n is the number of different terms, and m is the number of documents, in the collection. The VSM assumes that term vectors are pair-wise orthogonal. This assumption is very restrictive because the similarity between each document/query pair is only determined by the terms they have in common, not by the terms that are semantically similar in both.

There have been various extensions to the VSM. One example is Latent Semantic Analysis (LSA) [17], a method of word co-occurrence analysis to compute semantic vectors (context vectors) for words. LSA applies singular-value decomposition (SVD) to V (the term-document matrix) in order to construct context vectors. As a result, the dimension of the produced vector space will be significantly smaller by grouping together words that mean similar things; consequently the vectors that represent terms cannot be orthogonal. However, dimension reduction techniques such as SVD are expensive in terms of memory and processing time. As an alternative, there is a vector space methodology called Random Indexing (RI) [3], which represents an efficient, scalable, and incremental method for building context vectors, which express the distributional profile of linguistic terms.

RI overcomes the efficiency problems by incrementally accumulating k - dimensional index vectors into a context matrix R of order n x k, where $k \ll m$, but usually on the order of thousands. This is done in two steps: 1) A unique random representation known as index vector is assigned to each context (either document or word), consisting of a vector with a small number (ϵ) of non-zero elements, which are either +1 or -1, with equal amounts of both. For example, if index vectors have twenty non-zero elements in a 1024-dimensional vector space, they have ten +1s and ten -1s. Index vectors serve as indices or labels for words or documents; 2) Index vectors are used to produce context vectors by scanning through the text. Every time a target word (t) occurs in a context (c), the index vector of the context (ic) is added to the context vector of t (tc). Thus, the context vector of t is updated as: $tc += ic$.

In this way, R is a matrix of k-dimensional context vectors that are the sum of the terms' contexts. Notice that these steps will produce a standard term-document matrix V of order n x m if we use unary index vectors of the same dimensionality as the number of contexts. Such m-dimensional unary vectors would be orthogonal, whereas the k-dimensional random index vectors are only nearly orthogonal. However, Hecht-Nielsen [21] stated that there are many more nearly orthogonal directions in a high dimensional space than truly orthogonal directions, which means that context matrix R n x k will be an approximation of the term-document matrix F n x m.

The approximation is based on the Johnson-Lindenstrauss lemma [21], which states that if we project points in a vector space into a randomly selected subspace of sufficiently high dimensionality, the distances between the points are approximately preserved. Then, the dimensionality of a given matrix V can be reduced by projecting it through a matrix P.

$$R_{nxk} = V_{nxm} P_{mxk} \tag{1}$$

Random Indexing has several advantages: 1. It is incremental, which means that the context vectors can be used for similarity computations even after just a few documents have been processed; 2. It uses fixed dimensionality, which means that new data do not increase the dimensionality of the vectors; 3. It uses implicit dimensionality reduction, since dimensionality is much lower than the number of contexts in the data ($k \ll m$).

There are works that have validated the use of RI in text processing tasks: for example, Sahlgren & Karlgren [12] demonstrated that Random Indexing can be applied to parallel texts for automatic bilingual lexicon acquisition. Sahlgren & Cöster [3] used Random Indexing to carry out text categorization. This technique, as far as we know has not been used in IR, but similar techniques as SVD are well known and used in the area.

4 BoC Document Representation

BoC is a recent representation scheme introduced by Sahlgren & Cöster [3], which is based on the idea that the meaning of a document can be considered as

the union of the meanings of its terms. This is accomplished by generating term context vectors for each term within the document, and generating a document vector as the weighted sum of the term context vectors contained within that document. Thus, the m documents in a collection D are represented as:

$$\mathbf{d_i} = \sum_{j=1}^{s} h_{ji}\mathbf{g_j} \qquad i = 1, ..., m \tag{2}$$

where s is the number of terms in document d_i, $\mathbf{g_j}$ is the context vector of term j, and h_{ji} is the weight assigned to term j in the document i, according to the weighting scheme considered.

The context vectors used in BoC are generated using RI and 'Document Occurrence Representation' (DOR). DOR is based on the work of Lavelli et al. [13] and considers the meaning of a term as the bag of documents in which it occurs. When RI is used together with DOR, the term t is represented as a context vector:

$$\mathbf{t} = \sum_{k=1}^{u} \mathbf{b_k} \tag{3}$$

where u is the number of documents containing t, and $\mathbf{b_k}$ is the index vector of document k, then the contribution of document k to the specification of the semantics of term t. For instance, the context vector for a term t, which appears in the documents d_1 = [1, 0, -1, 0] and d_2 = [1, 0, 0, -1] would be [2, 0, -1, -1]. If the term t is encountered again in document d_1, the existing index vector of d_1 would be added one more time to the existing context vector to produce a new context vector for t of [3, 0, -2,-1]. Context vectors generated through this process are used to build document vectors as BoC. Thus, a document vector is the sum of the context vectors of its terms.

5 HRR Document Representation

In addition to BoC, we explore the use of syntactic structures (prepositional phrases such as *'in Asia')* to represent spatial relations and re-rank the retrieved documents. The traditional IR methods that include compound terms, extract and include them as new VSM terms [4], [5]. We explore a different representation of such structures, which uses a special kind of vector binding (called holographic reduced representations (HRRs) [2]) to reflect text structure and distribute syntactic information across the document representation. Fishbein, and Eliasmith have used the HRRs together with Random Indexing for text classification, where they have shown improvement under certain circumstances, having BoC as the baseline [1]. It is important to mention that up to now, we are not aware of other work that uses RI together with HRRs.

The Holographic Reduced Representation, HRR, was introduced by Plate [2] as a method for representing compositional structure in distributed representations. HRRs are vectors whose entries follow a normal distribution N(0,1/n).

They allow to express structure using a circular convolution operator to bind terms. This circular convolution operator ($\otimes$) binds two vectors $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ and $\mathbf{y} = (y_0, y_1, \ldots, y_{n-1})$ to produce $\mathbf{z} = (z_0, z_1, \ldots, z_{n-1})$ where $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$ is defined as:

$$z_i = \sum_{k=0}^{n-1} x_k \; y_{i-k} \qquad i = 0 \; to \; n-1 (\text{subscripts are modulo} - n) \tag{4}$$

Circular convolution is an operator which does not increase vector dimensionality, making it excellent for representing hierarchical structures. We adopt HRRs to build a text representation scheme in which spatial relations (SR) could be captured. Therefore, to define an HRR document representation, the following steps are done: a) Determine the index vectors for the vocabulary by adopting the random indexing method, as described earlier; b) Tag text of documents using a Name Entity Recognition System; c) Bind the $tf.idf$-weighted index vector of each location entity to its location role. This location role is an HRR which represents a preposition (i.e. *in, near, around, across*, etc.) extracted from the text considering the preposition preceding the location entity; d) Add the resulting HRRs (where the spatial relations are encoded) to obtain a single HRR vector; e) Multiply the resulting HRR by an attenuating factor α; f) Normalize the HRR obtained so far, to get the vector which represents the document. For example, when given a spatial relation: $R = in\ Asia$, R will be represented using the index vectors r_1 for Asia, where r_1 will be joined to its location role, an HRR, $role_1$ which represents the relation in. Then, the $in\ Asia$ vector will be:

$$\mathbf{R} = (\mathbf{role_1} \otimes \mathbf{r_1}) \tag{5}$$

Thus, given a document D, with spatial relations $in : t_{x1}, t_{y1}$, its normalized vector will be built as:

$$\mathbf{D} = \langle \alpha((\mathbf{role_1} \otimes \mathbf{t_{x1}}) + (\mathbf{role_1} \otimes \mathbf{t_{y1}}) \rangle \tag{6}$$

where α is a factor less than one intended to lower the impact of the coded relations. Queries are processed and represented in a similar way.

6 Experimental Setup

We used in our experiments Lemur[5]. The results produced by the VSM configured in Lemur were taken as our baseline.

Our experiments were conducted using the English document collection for the GeoCLEF track. This collection is composed of news articles taking 56, 472 from the Glasgow Herald (British) 1995 and 113, 005 from the LA Times (American) 1994 to total 169,477 news articles.

We worked with the queries of GeoCLEF 2007 and GeoCLEF 2008, a set of 50 queries (from number 51 to 100). These queries are described in three parts:

[5] http://www.lemurproject.org/

a) the main query or title; b) a brief description; and c) a narrative. We took the title and description for all our experiments, except for the query representations with HRR, where we also considered the narrative statement in order to have improved relations for representation. It is worth mentioning that Lemur results worsen when the narrative is included.

To investigate whether combining word-based and concept-based representations can be used to improve the GIR, we considered two phases. The aim of the first was to retrieve as many relevant documents as possible for a given query, whereas the purpose of the second was to improve the final ranking of the retrieved documents by applying BoC and HRR representations.

Lemur was used to process the 169,477 documents, first with the queries for 2007 and then with the queries for 2008. Thereafter, only the top 1000 documents ranked by the VSM were selected for each query. These sub-collections were processed to generate the BoC representations of its documents and queries. BoC representations were generated by first stemming all words in the sub-collections, using the Porter stemmer. We then used Random Indexing to produce context vectors for the given sub-collection. The dimensionality of the context vectors was fixed at 4096. The index vectors were generated with 10 +1s and 10 -1s, distributed over the 4096 dimensions. This vector dimension and density were empirically determined. These context vectors were then $tf.idf$-weighted and added up for each document and query, as described earlier to produce BoC representations.

On the other hand, HRRs were generated by firstly tagging all sub-collections with the Named Entity Recognition System of Stanford University[6]. Afterwards, the single word locations preceded by the preposition *in* were extracted. This restriction was taken after analyzing the queries for each year and realizing that only about 12% of them had a different spatial relation. HRRs for documents and queries were then produced by generating a 4096-HRR to represent the *in* relation. The *in* HRR vector was then bound to the index vector of the identified locations by a Fast Fourier Transform implementation of circular convolution, $tf.idf$-weighted, added, and multiplied by $\alpha = 1/6$ to represent each document, as described earlier to generate spatial relations representations.

Finally, the evaluation of the results after re-ranking the documents was carried out with the Mean Average Precision (MAP).

7 Results

We consider two experiments: a) The aim of the first was to prove that incorporating context information and syntactic structure for re-ranking documents in GIR could improve precision (i.e. to explore the use of BoC and HRR representations) b) The objective of the second was to compare our strategies against a traditional re-ranking mechanism known as Pseudo Relevance Feedback (PRF).

First Experiment. Table 1 compares Lemur results, with the results produced by adding the Lemur similarity values with its corresponding values from BoC to

[6] http://nlp.stanford.edu/software/CRF-NER.shtml

Table 1. MAP results for Geo-CLEF collection (2007 - 2008)

	Lemur	Lemur-BoC	%Diff	Lemur-BoC-HRR	% Diff
2007	0.1832	0.2079	13.48	0.2085	13.81
2008	0.2445	0.2619	7.12	0.2628	7.48

produce Lemur-BoC, which is a new list re-ranked according to the new values. Then the same process as described above was followed, but now adding Lemur-BoC values to HRR values to produce Lemur-BoC-HRR. We only considered the set of supported queries, that is, the queries that have at least one relevant document: 22 queries in 2007 and 24 in 2008. Notice how MAP is incremented in a constant way, always at above 7%.

From the queries considered in 2007, 1 query kept the same MAP produced by Lemur after adding BoC. The MAP of 5 queries decreased. Positively, there are 16 queries improved by BoC. The favorable percentages of improvement for 10 queries are observed in Table 2 above the 14%.

When HRRs were added to Lemur-BoC, only the query 64 that was not improved by BoC (and in consequence, not in Table 2) was affected. This query had a percentage of change equal to -4.35%, which was raised to 30.43% by the representation of its 5 spatial relations. From the queries shown in Table 2, the unaffected queries have none or one spatial relation, while the queries enhanced by adding the HRRs have on average 4.

We found that HRRs improve precision when there are distinctive and specific spatial relations, for example: *in Finland* instead of *in northern Europe*. Therefore when geographical information given is more precise, HRRs help to achieve improved effectiveness. However, when the number of retrieved relevant documents

Table 2. MAP for query improvement by BoC in 2007 and 2008 and their spatial relations

	Qry-ID	Lemur	Lemur - BoC	% Diff	SR	Lemur-BoC-HRR	%Diff. additional
Results 2007	52	0.0022	0.0038	72.73	0	0.0038	0.00
	57	0.204	0.2473	21.23	6	0.2577	4.21
	58	0.0197	0.0268	36.04	0	0.0268	0.00
	60	0.0022	0.0397	1704.55	1	0.0397	0.00
	61	0.0959	0.1321	37.75	1	0.1318	-0.23
	67	0.2569	0.2950	14.83	0	0.2950	0.00
	69	0.0701	0.0964	37.52	1	0.0963	-0.14
	70	0.043	0.0509	18.37	0	0.0509	0.00
	72	0.4859	0.6179	27.17	1	0.6179	0.00
	75	0.3522	0.4580	30.04	2	0.4612	0.70
Results 2008	76	0.44	0.4857	10.39	12	0.5000	2.94
	80	0.2518	0.2555	1.47	1	0.2555	0.00
	82	0.0005	0.0015	200.00	3	0.0018	20.00
	84	0.1385	0.2183	57.62	0	0.2183	0.00
	85	0.4554	0.4767	4.68	0	0.4767	0.00
	86	0.0592	0.1101	85.98	2	0.1130	2.63
	91	0.0625	0.1667	166.72	1	0.1667	0.00
	93	0.7375	0.8340	13.08	1	0.8340	0.00
	95	0.491	0.5320	8.41	6	0.5337	0.26
	96	0.2232	0.2418	8.33	11	0.2454	1.49

is low with few relations to compare, it is difficult to affect the ranking with the HRRs.

In 2008, 3 queries kept the same MAP produced by Lemur after adding BoC. The MAP of 9 queries decreased and 12 queries improved. Table 2 shows 10 queries improved by BoC where favorable percentages of improvement are depicted. From these 10 queries, those that were improved after adding the HRRs, have at least 2 spatial relations. Our conclusion is that the relative small contribution to improve precision demonstrated by HRR is due to the limited amount of spatial relations appearing in the set of queries used. We believe that the higher the number of spatial relations to be represented, the greater the contribution of this representation.

We perform a paired t-student test to measure the statistical significance of our MAP results. The MAP differences for GeoCLEF 2007 resulted significant in a confidence interval of 95% for both Lemur-BoC and Lemur-BoC-HRR; however the results are below the median of the year (0.2097) by 0.57%. In this year, the top system at CLEF reached a MAP of 0.2859 [9]. However, it used a very complex configuration and several external resources (four Geographical Gazetteers, a Feature Type Thesaurus to categorize geo-terms and a Shape Toolbox a database, which contains a "shape file" available for each country).

The MAP improvement for 2008 is not statistically significant. Even so, the MAP median of the participants in Geo-CLEF 2008 was of 0.2370 [15], which is 6.45% lower than that generated by our proposal. This year the team at the top obtained a MAP of 0.3040 [6]. They used two ontologies constructed manually, employing information from narratives. In addition they used Wikipedia in the retrieval process. In contrast we do not use any complex external resource.

Second Experiment. Finally, we compare the Lemur-BoC-HRR results with a traditional re-ranking method known as Pseudo Relevance Feedback (PRF). In order to apply this approach, we used the VSM, representing queries and documents as *tf-idf* vectors, and computing similarity with the cosine function. PRF treats the n top ranked documents as true relevant documents for a given query, then queries are expanded by adding the k words selected from the n top documents, and then a second IR process is done with the expanded query. Table 3 presents results (also for queries with relevant documents) when the top 2 and 5 documents are taken to extract 5, 10, and 15 words. Query texts are built from title and description fields. The values that improve Lemur MAP are

Table 3. Difference between PRF MAP and Lemur-BoC-HRR MAP

	Lemur-BoC-HRR	PRF with 2 documents			PRF with 5 documents		
		5 terms	10 terms	15 terms	5 terms	10 terms	15 terms
GeoCLEF 2007	*0.2085*	**0.1925**	0.1617	0.1533	**0.1963**	0.1703	0.1593
% Difference		8.31	28.94	36.01	6.21	22.43	30.89
GeoCLEF 2008	*0.2628*	**0.2539**	**0.2596**	**0.2505**	0.2306	0.2242	0.2101
% Difference		3.51	1.23	4.91	13.96	17.22	25.08

depicted in bold and those obtained with our proposal in italics. The difference in MAP between PRF technique and our Lemur-BoC-HRR proposal is about 6.21% or higher in favor of our method in 2007 and 1.23% or higher in 2008.

8 Conclusion and Future Work

In this paper, we have presented two document representations for re-ranking documents and improving precision for GIR. RI was used to build context vectors to create BoC representations, which capture context information. It also defines index vectors used in the HRR representations. When working with RI, the appropriate selection of the values for vector length and vector density is an open research topic. Our results have been compared with the VSM in its Lemur implementation. They have showed that: i) BoC can improve the initial ranker. ii) HRR representation improved the ranking of queries. However, its utility could not be totally verified because of the lack of spatial relations to be represented; ii) we foresee that when more relations are added to the HRRs, a better ranking is achieved. It should be noted that in the experiments conducted, only one type of spatial relation (*in*) was considered: we think if more types of relations (*near, around, across, far*, etc.) are added as long as they are present in the queries; it could lead to improved results; iii) comparing our method against PRF produces higher scores for this new method. Therefore, the overall results demonstrate that our approach is appropriate for re-ranking documents in GIR.

We will continue working with other collections where queries have not only spatial relations but other syntactic relations (i.e. compound nouns, verb-subject) which could be represented and together with the context information, allow us to explore in-depth the usefulness of the proposed representations as a mechanism for re-ranking documents to improve precision.

References

1. Fishbein, J.M., Eliasmith, C.: Integrating structure and meaning: A new method for encoding structure for text classification. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 514–521. Springer, Heidelberg (2008)
2. Plate, T.A.: Holographic Reduced Representation: Distributed representation for cognitive structures. CSLI Publications, Stanford (2003)
3. Sahlgren, M., Cöster, R.: Using Bag-of-Concepts to Improve the Performance of Support Vector Machines in Text Categorization. In: Procs. of the 20th International Conference on Computational Linguistics, pp. 487–493 (2004)
4. Mitra, M., Buckley, C., Singhal, A., Cardie, C.: An Analysis of Statistical and Syntactic Phrases. In: Procs. 5th International Conference of RIAO 1997, pp. 200–214 (1997)
5. Evans, D., Zhai, C.: Noun-phrase Analysis in Unrestricted Text for Information Retrieval. In: Procs. of the 34th Annual Meeting on ACL, pp. 17–24 (1996)
6. Wang, R., Neumann, G.: Ontology-based query construction for Geoclef. In: Working notes for the CLEF Workshop, Aarhus, Denmark (2008)

7. Martinis, B., Cardoso, N., Chavez, M.S., Andrade, L., Silva, M.J.: The University of Lisbon at Geoclef 2006. In: Working notes for the CLEF Workshop, Spain (2006)
8. Larson, R.R.: Cheshire at Geoclef 2008: Text and fusion approaches for GIR. In: Working notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
9. Ferrés, D., Rodríguez, H.: TLAP at GeoCLEF 2007: Using Terries with Geographic Knowledge Filtering. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 830–833. Springer, Heidelberg (2008)
10. Larson R.R.: Cheshire II at GEOCLEF 2005: Fusion and query expansion for GIR. In: Working notes for the CLEF 2005 Workshop, Wien, Austria (2005)
11. Villatoro-Tello, E., Montes-y-Gómez, M., Villaseñor-Pineda, L.: INAOE at GEOCLEF 2008: A ranking approach based on sample documents. In: Working notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
12. Sahlgren, M., Karlgren, J.: Automatic bilingual lexicon acquisition using Random Indexing of parallel corpora. Journal of Natural Language Engineering Special Issue on Parallel Texts 11(3), 327–341 (2005)
13. Lavelli, A., Sebastiani, F., Zanoli, R.: Distributional term representations: an experimental comparison. In: CIKM 2004: Procs. of the Thirteenth ACM Conference on Information and Knowledge Management, pp. 615–624. ACM Press, New York (2004)
14. Cross-lingual evaluation forum (2009), http://www.clef-campaign.org/
15. Mandl T., Carvalho P., Gey F., Larson R., Santos D., Womser-Hacker C., Di Nunzio G., Ferro N.: Geoclef 2008: the CLEF 2008 Track Overview. In: Working notes for the CLEF Workshop, Aarhus, Denmark (2008).
16. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11), 613–620 (1975)
17. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. Journal of the ASIS 41, 391–407 (1990)
18. Henrich, A., Lüdecke, V.: Characteristics of Geographic Information needs. In: Procs. of Workshop on Geographic Information Retrieval, Lisbon, Portugal. ACM Press, New York (2007)
19. Andrade, L., Silva, M.J.: Relevance ranking for geographic IR. In: Procs. of 3rd Workshop on Geographic Information Retrieval, SIGIR 2006. ACM Press, New York (2006)
20. Kanerva, P., Kristoferson, J., Anders Holst, A.: Random indexing of text samples for latent semantic analysis. In: Procs. of the 22nd Annual Conf. of the Cognitive Sc. Society, USA (2000)
21. Sahlgren, M.: An introduction to random indexing. In: Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, Copenhagen, Denmark (2005)

Using Machine Translation Systems to Expand a Corpus in Textual Entailment

Julio J. Castillo

National University of Cordoba-FaMAF, Cordoba, Argentina
National Technological University-FRC, Cordoba, Argentina

Abstract. This paper explores how to increase the size of Textual Entailment Corpus by using Machine Translation systems to generate additional $\langle t, h \rangle$ pairs. We also analyze the theoretical upper bound of a Corpus expanded by machine translation systems, and propose how it computes the confidence of a classification translator-based RTE system. At the end, we show an algorithm to expand the corpus size using Translator engines and we provide some results over a real RTE system.

Keywords: textual entailment, machine translation system, double translation process.

1 Introduction

The Recognizing Textual Entailment (RTE) task is defined as a directional relationship between a pair of text fragments or sentences, called the "text" (T), and the "hypothesis" (H). Thus, we say that "T entails H", if a human reads T would infer that H is most likely true.

Machine learning algorithms were widely used for the task of recognizing textual entailment [1], [2], [3] in the past RTE Challenges. Some authors [4] showed how the accuracy increases when we add more training examples, and other authors holds the necessity of larger corpus [5]. In any case, a larger corpus enables a more detailed analysis of the problem domain and will let us build more accurate classifiers.

In this paper we show how a machine translation system could increase the size of a RTE Corpus, and we also suggest how a translator can be used as a tool to help us with classification of new (unknowns) $\langle t, h \rangle$ pairs.

The remainder of the paper is organized as follows: Section 2 describes one approach driven by Machine Translation Systems and provides an analysis about the possible increasing size of the Corpus, with us proposing a confidence measure for such approach, whereas Section 3 shows experimental evaluation and discussion of the results.

Finally, Section 4 summarizes the conclusions and lines for future work.

2 Machine Translation Approach

In this section, we propose to use Machine Translation System to expand the current RTE-Corpus sizes.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 97–102, 2010.

First, we define "double translation process" as the process of starting with a String (in English), translating it to another language, for example Spanish, and backing it forward to the English language (source).

Thus, our motivation is based on the fact that we could use a double translation process to produce equivalents Texts and Hypothesis, and so these new pairs can be taken as training set. Also, we suggest how a translator can be used as a tool to help us with classification of new $\langle t, h \rangle$ pairs.

2.1 Double Translation Process

Double translation process can be defined as the process of starting with an S (String in English), translating it to a foreign language F(S), for example Spanish, and backing it forward to the English source language $F^{-1}(S)$. Thus, the observation of that double translation process can increase the Corpus size and also can be generalized using N-Translators engine.

It is important to note that the "quality" of the translation is given by the Machine Translation System, and we will suppose that the sense of the sentence should not be modified by the Translator. This, indeed, is the situation almost for the majority of the cases in our first experiments (see Section 2.2). Bellow, we provide a theoretical justification of the increment of the corpus size with n-pairs using k-translators which is $O(n * k^2)$.

Notation: C is a RTE Corpus which consists of $\langle t, h \rangle$ pairs. C_q is the increased size of the Corpus C using q–Translators.

$$\text{Notation: } \langle t, h \rangle \equiv t \rightarrow h$$

Define:

$$Tr : String \rightarrow String$$
$$t \rightarrow Tr(t)$$
$$Tr(t) = DoubleTranslationOfTheTrTranslator$$

where: t and Tr(t) are in English.

Lemma: Given Tr1, Tr2,…,Trk translators and C a RTE Corpus with n- $\langle t, h \rangle$ pairs. If

$$Tr_i(t_p) \neq Tr_j(h_p) \quad \forall i, j \wedge i \neq j \wedge i \in \{1,...,k\} \wedge p \in \{1,..n\}$$

then $C_k = (k+1)^2 * n$.

Proof: By structural induction on K.

As a practical result, by using one translator over only one RTE dataset of 800 pairs, we could obtain up to 3200 pairs in total, and using two translators we could obtain a new dataset with 7200 pairs (upper bound).

2.2 Other Uses of Machine Translation in RTE Systems

Machine Translation can be used as a feature in a machine learning algorithm [6]. Indeed, by using a MT system it is possible to reduce the complexity of some of the sentences and by this way, RTE task will be easier. We addressed several simple experiments using Machine Translation engines, and we provide some examples below.

One original $\langle t, h \rangle$ pair of RTE3 development set is:

T: *A leading human rights group on Wednesday identified Poland and Romania as the likely locations in eastern Europe of secret prisons where al-Qaeda suspects are interrogated by the Central Intelligence Agency.*

H: *CIA secret prisons were located in Eastern Europe.*

Translated pair using Microsoft Bing Translator:

T: *A group of human rights on Wednesday had identified Poland and Romania as likely locations in Eastern Europe of secret prisons where suspects of Al Qaida are interrogated by the Central Intelligence Agency.*

H: *Secret CIA prisons were in Eastern Europe.*

Translated pair using Google Translator:

T: *A prominent human rights group on Wednesday identified Poland and Romania as the likely locations in eastern Europe of secret prisons where al-Qaeda suspects are interrogated by the Central Intelligence Agency.*

H: *the secret CIA prisons located in Eastern Europe.*

Translated pair using Yahoo Babel Fish Translator:

T: *A main group of human rights Wednesday identified Poland and Rumania like the probable locations in Eastern Europe of secret prisons where the company the suspects of al-Qaeda interrogate.*

H: *The secret prisons of the company were located in Eastern Europe.*

In order to move the RTE task towards more realistic application scenarios, this year in the TAC RTE5 Challenge, the texts come from a variety of sources and include typographical errors and ungrammatical sentences. In this context, the translation can help with this objective. In the previous example, "eastern" is a grammatical error on the original corpus but using Bing Translator this error was fixed. Also, we can produce some interesting variation such as "*al-Qaeda*" and "*Al Qaida*".

In the last pair, by using BabelFish the "CIA" was changed for "company", which is an error of the translator in the context of this pair. It seems an error of Babel Fish resolving acronyms.

Another use of Translator is to provide synonyms and expression with the same meaning. An additional example is given bellow. Again, the following pair belongs to the RTE3 development set.

The source pair 170 is (entailment = Yes):

T: The man known as just the "Piano Man" has left the hospital and has returned home to his native Germany. According to British tabloids, the man, after losing his job in Paris, travelled to the UK through the Channel Tunnel.

H: The "Piano Man" came from Germany.
Translated pair using Microsoft Bing Translator:

T: The man known an as the "Piano Man" has left the hospital and has returned to his native Germany. According to British tabloids man after losing her job in Paris, traveled to the United Kingdom on the channel tunnel.

H: "Piano man" came from Germany.

In this example, we see how "*UK*" was translated to "*United Kingdom*" (acronyms resolution), and also we see interesting variations as "traveled" and "travelled". Thus, it seems that by using a MT engine it is possible to improve the semantic resources of RTE Systems.

2.3 Confidence of Double Translation Process

The double translation process can be used in the production step when addressing machine learning algorithms, or otherwise in testing stage in other systems.

By this way, we can define for a test set $\langle t_i, h_i \rangle$ pair.

$$RTE\ (T, H) = \begin{cases} 1, & si\ T \Rightarrow H \\ 0, & otherwise \end{cases}$$

In this case, RTE is a result classification of a system for Recognizing Textual Entailment.
Then, we can define Confidence as:

$$Confidence(T, H) = \frac{\sum_{i=1}^{n} RTE(Tr_i(T), Tr_i(H))}{n}$$

Thus, we can choose a threshold and only accept as a valid entailment a pair that outperforms this threshold.

Additionally, it is possible to prove that $C_k = (k+1)^2 * n$ is the upper-bound for a double translation process and this bound only could be reached when the predicate $Tr_i(t_p) \neq Tr_j(h_p) \forall i, j \wedge i \neq j \wedge i \in \{1,...,k\} \wedge p \in \{1,..n\}$ holds.

Generally, H is very simple. For this reason, a double translation process could not affect the Hypothesis. On the other hand, since T (Text) is complex, we expect that every translator engine will return a different result.

3 Experimental Evaluation and Discussion of the Results

We show the following algorithm in order to obtain additional $\langle t, h \rangle$ pairs from a given Corpus:

1. Start with a RTE-x Corpus, with $|C| = n$
2. For each $\langle t_i, h_i \rangle \quad \wedge i \in \{1,...,n\}$
3. For each Translator Tr1, Tr2,…,Trk. If

$$Tr_j(t_i) \neq t_i \wedge Tr_j(h_i) \neq h_i \ \forall j \in \{1,..,k\} \rightarrow Add(Tr_j(t_i), Tr_j(h_i)) \ to \ C_{new}$$

Where: C_{new} is the new Corpus obtained as the union between C and the new outputs pairs of the algorithm.

In order to test our claim over a real RTE system we performed some experiments using our RTE system [3], but without using the NER filter. This RTE system is based on a machine learning approach that produces feature vectors for RTE3, RTE4, and RTE5. The chosen features quantify lexical, syntactic and semantic level by matching between texts and hypothesis sentences. Thus, we generated a feature vector with the following components for both Text and Hypothesis: Levenshtein distance, a lexical distance based on Levenshtein, a semantic similarity measure based on Wordnet, and LCS (longest common substring) metric.

We use the following two classifiers to learn every development set: Support Vector Machine, and Multilayer Perceptron (MLP) and we choose Spanish as intermediate language. First, we started with RTE3 datasets applying the algorithm proposed using only one Machine Translation System (Microsoft Bing Translator) in order to generate additional pairs, which was named as RTE3-Bing. Secondly, we split this new dataset in 200, 400, 600 and 800 pairs respectively. Finally, we tested these training sets over RTE5 development set in two-way task. We summarized the results in the following table:

Table 1. Results of two-way classification task

Training set	SVM Classifier	MLP Classifier
RTE3	55.5	56
RTE3-Bing	56.5	58
RTE3+200pairs-RTE3-Bing	55.67	56.5
RTE3+400pairs-RTE3-Bing	55.83	58.83
RTE3+600pairs-RTE3-Bing	56.5	57.67
RTE3+RTE3-Bing	56	59.33

Interestingly, a not statistical significant different was obtained between RTE3 and RTE3-Bing using SVM or MLP as learning algorithm.

One important point in these experiments is that adding more training set obtained with our algorithm, does not decrease the performance with neither combination of learning algorithm and training sets.

Finally, the best performance of our system was achieved with Machine Learning Algorithm with RTE3+RTE3-Bing dataset and it was obtained an interesting increase but not statistically significant of 3.33% (accuracy).

However, additional evidence is needed in order to support this claims, but it seems promising. Also, it is important to note that translation web services as Google

Translator or Microsoft Bing is frequently updated. Therefore, the result of the translation could not be the same at different times, and so we would expect better results.

4 Conclusions and Future Work

In this work we propose the use of Machine Translation systems as a way to increase the corpus sizes. We also show the maximum size which can be yield and we present an algorithm to increase training sets.

We concluded that for our algorithm, Microsoft and Google MT are more useful that Yahoo Babelfish MT, and also we note that these results are strong dependent of the RTE system architecture.

However, further analysis is required to determine the impact of Machine Translation in Textual Entailment Systems by using others RTE systems.

Future work will be oriented to explore more deeply how Machine Translation could improve the accuracy of the RTE Systems, and to test over different datasets and RTE systems available.

Finally, we will test the double translation process but passing through Spanish, Portuguese, Dutch, and Russian as intermediate language, and assessing the improvement that they can yield.

References

1. Marneffe, M., MacCartney, B., Grenager, T., Cer, D., Rafferty, A., Manning, C.: Learning to distinguish valid textual entailments. In: RTE2 Challenge, Italy (2006)
2. Zanzotto, F., Pennacchiotti, M., Moschitti, A.: Shallow Semantics in Fast Textual Entailment Rule Learners. In: RTE3, Prague (2007)
3. Castillo, J.: Recognizing Textual Entailment: Experiments with Machine Learning Algorithms and RTE Corpora. In: Cicling 2010, Iaşi, Romania (2009)
4. Inkpen, D., Kipp, D., Nastase, V.: Machine Learning Experiments for Textual Entailment. In: RTE2 Challenge, Venice, Italy (2006)
5. Newman, E., Stokes, N., Dunnion, J., Carthy, J.: UCD IIRG Approach to the Textual Entailment Challenge. In: PASCAL. Proc. of the First Challenge Workshop. Recognizing Textual Entailment (2005)
6. Agichtein, E., Askew, W., Liu, Y.: Combining Lexical, Syntactic, and Semantic Evidence for Textual Entailment Classification. In: TAC 2008, Gaithersburg, Maryland, USA (2008)
7. Bentivogli, L., Dagan, I., Dang, H., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of Textual Analysis Conference, NIST, Maryland, USA (2009)
8. Dolan, B., Quirk, C., Brockett, C.: Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, p. 350 (2004)
9. Castillo, J.: A Machine Learning Approach for Recognizing Textual Entailment of the Spanish. In: North American Chapter of ACL (2010)
10. Vanderwende, L., Dolan, W.B.: What syntax can contribute in entailment task. Springer, Heidelberg (2006)
11. Dagan, I., Dolan, B., Magnini, B., Roth, D.: Recognizing textual entailment: Rational, evaluation and approaches. Natural Language Engineering 15(4), i–xvii (2009)

Frames in Formal Semantics

Robin Cooper

Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
Box 200
S-405 30 Göteborg, Sweden
http://www.ling.gu.se/~cooper/

Abstract. In his classic paper on frame semantics, Charles Fillmore says that it comes from traditions of empirical semantics rather than formal semantics. In this paper we will try to draw a closer connection between empirical and formal semantics and suggest that a notion of frame closely related to that found in FrameNet can be usefully exploited in formal semantics based on a particular type theory with records (TTR). We will first show how frames of this kind can be used to give a compositional semantics for verbs relating to Reichenbach's analysis of tense using speech, reference and event time. We will then revisit an old puzzle from Montague semantics relating to temperature and price. We will relate our solution to this puzzle to Fernando's string theory of events. Finally, we will consider some consequences of our analysis for the way in which agents acquire and modify word meaning as a result of exposure to linguistic input.

Keywords: frame semantics, type theory, lexical semantics, mathematical modelling.

1 Introduction

In his classic paper on frame semantics, Fillmore [12] says:

> Frame semantics comes out of traditions of empirical semantics rather than formal semantics. It is most akin to ethnographic semantics, the work of the anthropologist who moves into an alien culture and asks such questions as, 'What categories of experience are encoded by the members of this speech community through the linguistic choices that they make when they talk?' A frame semantics outlook is not (or is not necessarily) incompatible with work and results in formal semantics; but it differs importantly from formal semantics in emphasizing the continuities, rather than the discontinuities, between language and experience. The ideas I will be presenting in this paper represent not so much a genuine theory of empirical semantics as a set of warnings about the kinds of problems such a theory will have to deal with. If we wish, we can think of the remarks I make as 'pre-formal' rather than 'non-formalist'; I claim to be listing, and as well as I can to be describing, phenomena

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 103–114, 2010.

which must be well understood and carefully described before serious formal theorizing about them can become possible.

In this paper, we will make a connection between formal semantics and frame semantics by importing into our semantic analysis objects which are related to the frames of FrameNet.[1] Our way of doing this will be different from, for example, [1]. An important part of our proposal will be that we introduce semantic objects corresponding to frames and that these objects can serve as the arguments to predicates. We will use record types as defined in TTR (type theory with records, [2,3,5,13]) to characterize our frames. The advantage of records is that they are objects with a structure like attribute value matrices as used in linguistics. Labels (corresponding to attributes) in records allow us to access and keep track of parameters defined within semantic objects. This is in marked contrast to classical model theoretic semantics where semantic objects are either atoms or unstructured sets and functions. We will first give a brief intuitive introduction to TTR and show how it can be used to represent frames (Sect. 2). We will then show how we propose to represent the contents of verbs in a compositional semantics (Sect. 3). The use of frames here leads us naturally from the Priorean tense operators used by Montague to the Reichenbachian account of tense [22] preferred by most linguists working on tense and aspect which involves what we will think of as parameters for speech time, event time and reference time. The use of frames also leads us to a particular view of Partee's puzzle about temperature and price first discussed in [16] (PTQ, reprinted as Chap. 8 of [17]). We will discuss this in Sect. 4. Our solution to this puzzle relates to Fernando's ([9,11]) theory of events as strings of frames which we discuss in Sect. 5. Finally (Sect. 6), we will consider how our proposal can be used to talk about how agents can modify word meaning by adjusting the parameters of word contents. This relates to a view of word meaning as being in a constant state of flux as we adapt words to describe new situations and concepts. In Sect. 7 we draw some conclusions.

2 Using TTR to Represent Frames

Consider the frame `Ambient_temperature` defined in the Berkeley FrameNet[2] by "The Temperature in a certain environment, determined by Time and Place, is specified". Its core frame elements are given in (1).

(1) **Attribute.** The temperature feature of the weather
Degree. A modifier expressing the deviation of the Temperature from the norm
Place. The Place where it is a certain Temperature
Temperature. A quantity or other characterization of the Temperature of the environment
Time. The Time during which an ambient environment has a particular Temperature

[1] http://framenet.icsi.berkeley.edu/
[2] Accessed 25th Oct, 2009.

To make things of a manageable size we will not include all the frame elements in our representation of this frame. (We have also changed the names of the frame elements to suit our own purposes.) We will say that an ambient temperature frame is a record of type (2).

$$(2) \quad \begin{bmatrix} \text{x} & : \mathit{Ind} \\ \text{e-time} & : \mathit{Time} \\ \text{e-location} & : \mathit{Loc} \\ \text{c}_{\text{temp\_at\_in}} & : \text{temp\_at\_in(e-time, e-location, x)} \end{bmatrix}$$

We will call this type *AmbTemp*. It is a set of four fields each consisting of a *label* (to the left of the colon) and a type (to the right of the colon). A record of type *AmbTemp* will meet the following two conditions:

- it will contain *at least* fields with the same labels as the type (it may contain more)
- each field in the record with the same label as a field in the record type will contain an object of the type in the corresponding field of the record type. (Any additional fields with different labels to those in the record type may contain objects of any type.)

Types constructed with predicates such as 'temp_at_in' have a special status in that they can be *dependent*. In (2) the type in the field labelled '$\text{c}_{\text{temp\_at\_in}}$' depends on what you choose for the other three fields in the frame. Intuitively, we can think of such types formed with a predicate like 'temp_at_in' as types of objects which prove a proposition. What objects you take to belong to these types depends on what kind of theory of the world you have or what kind of application you want to use your type theory for. Candidates would be events, states or, in this case, thermometer or sensor readings. Types constructed with predicates are also used in representing the contents of verbs as we will see in Sect. 3.

3 A TTR Approach to Verbs in Compositional Semantics

Consider an intransitive verb such as *run*. The simplest way to think of this is as corresponding to a predicate of individuals. Thus (3) would represent the type of events or situations where the individual a runs.

(3) $\text{run}(a)$

However, as anybody who has thought about tense and aspect knows, we need to get time into the picture somewhere. If you look up *run* on FrameNet[3] you will find that on one of its readings it is associated with the frame `Self_motion`. Like many other frames in FrameNet this has a frame element `Time` which in

[3] Accessed 1st April, 2010.

this frame is explained as "The time when the motion occurs". This is what Reichenbach [22] called more generally *event time* and we will use the label 'e-time'. We will add an additional argument for a time to the predicate and create a frame-type (4).[4]

$$(4)\quad \begin{bmatrix} \text{e-time} : \mathit{TimeInt} \\ \text{c}_{\text{run}} \quad : \text{run}(a,\text{e-time}) \end{bmatrix}$$

For the type (4) to be non-empty it is required that there be some time interval at which a runs. We use *TimeInt* as an abbreviation for the type of time intervals, (5).

$$(5)\quad \begin{bmatrix} \text{start} : \mathit{Time} \\ \text{end} \quad : \mathit{Time} \\ \text{c} \quad\quad : \text{start}<\text{end} \end{bmatrix}$$

No constraints are placed on when that time interval in (4) should be. Thus this frame type corresponds to a "tenseless proposition", something that is not available in the Priorean setup [18,19] that Montague employs where logical formulae without a tense operator correspond to a present tense interpretation. In order to be able to add tense to this we need to relate the event time to another time interval, normally the time which Reichenbach calls the speech time.[5] A past tense type anchored to a time interval ι is represented in (6).

$$(6)\quad \begin{bmatrix} \text{e-time} : \mathit{TimeInt} \\ \text{c}_{\text{tns}} \quad : \text{e-time.end}< \iota.\text{start} \end{bmatrix}$$

This requires that the end of the event time interval has to precede that start of the speech time interval. In order for a past-tense sentence *a ran* to be true we would need to find an object of both types (4) and (6). This is equivalent to requiring that there is an object in the result of merging the two types given in (7).

$$(7)\quad \begin{bmatrix} \text{e-time} : \mathit{TimeInt} \\ \text{c}_{\text{tns}} \quad : \text{e-time.end}< \iota.\text{start} \\ \text{c}_{\text{run}} \quad : \text{run}(a,\text{e-time}) \end{bmatrix}$$

Suppose that we have an utterance u, that is, a speech event of type (8).

$$(8)\quad \begin{bmatrix} \text{phon} \quad : \text{“a”}^\frown\text{“ran”} \\ \text{s-time} : \mathit{TimeInt} \\ \text{c}_{\text{utt}} \quad : \text{uttered(phon,s-time)} \end{bmatrix}$$

[4] Of course, we are ignoring many other frame elements which occur in FrameNet's `Self_motion` which could be added to obtain a more detailed semantic analysis.

[5] Uses of historic present tense provide examples where the tense is anchored to a time other than the speech time.

where "a"⌢"ran" is the type of strings of an utterance of *a* concatenated with an utterance of *ran*. Then we can say that the speech time interval ι in (7) is u.s-time. That is, the past tense constraint requires that the event happened before the start of the speech event. In a complete treatment both the type of the speech event (8) and the content (7) would be packeted together in a single sign type together with more information about syntax, HPSG style (see [4] for a preliminary indication of how this would look).

(7) is a type which is the content of an utterance of the sentence *a ran*. In order to obtain the content of the verb *ran* we need to create a function which abstracts over the individual *a*. Because frames will play an important role as arguments to predicates below we will not abstract over individuals but rather over frames containing individuals. The content of the verb *ran* will be (9).

$$(9)\quad \lambda r{:}\left[\mathrm{x}{:}\mathit{Ind}\right]\left(\begin{bmatrix} \text{e-time} : \mathit{TimeInt} \\ \mathrm{c_{tns}} \quad : \text{e-time.end}< \iota.\text{start} \\ \mathrm{c_{run}} \quad : \text{run}(r.\text{x,e-time}) \end{bmatrix}\right)$$

4 The Puzzle about Temperature and Prices

Montague [16] introduces a puzzle presented to him by Barbara Partee:

> From the premises **the temperature is ninety** and **the temperature rises**, the conclusion **ninety rises** would appear to follow by normal principles of logic; yet there are occasions on which both premises are true, but none on which the conclusion is.

Exactly similar remarks can be made substituting *price* for *temperature*. Montague's solution to this puzzle in [16] was to analyze *temperature*, *price* and *rise* not as predicates of individuals as one might expect but as predicates of individual concepts. For Montague individual concepts were modelled as functions from possible worlds and times to individuals. To say that *rise* holds of an individual concept does not entail that *rise* holds of the individual that the concepts finds at a given world and time. Our strategy is closely related to Montague's. However, instead of using individual concepts we will use frames. By interpreting *rises* as a predicate of frames, for example, of type *AmbTemp* as given in (2) we obtain a solution to this puzzle.

$$(10)\quad \lambda r{:}\left[\mathrm{x}{:}\mathit{Ind}\right]\left(\begin{bmatrix} \text{e-time} : \mathit{TimeInt} \\ \mathrm{c_{tns}} \quad : \text{e-time}= \iota \\ \mathrm{c_{run}} \quad : \text{rise}(r,\text{e-time}) \end{bmatrix}\right)$$

Note that a crucial difference between (9) and (10) is that the first argument to the predicate 'rise' is the complete frame r rather than the value of the x field which is used for 'run'. Thus it will not follow that the value of the x field (i.e. 90 in Montague's example) is rising. While there is a difference in the type of

the argument to the predicates (a record as opposed to an individual), the type of the complete verb content is the same: $[\text{x}{:}\mathit{Ind}] \rightarrow \mathit{RecType}$, that is, a function from records of type $[\text{x}{:}\mathit{Ind}]$ to record types. This ability to use different types internally but still have the same overall type for the content of the word is convenient for compositional semantics.

But now the question arises: what can it mean for a frame to rise?

5 Fernando's String Theory of Events

In an important series of papers including [8,9,10,11], Fernando introduces a finite state approach to event analysis where events can be seen as strings of punctual observations corresponding to the kind of sampling we are familiar with from audio technology and digitization processing in speech recognition. When talking about the intuition behind this analysis Fernando sometimes refers to strings of frames in a movie (e.g. in [10]). But in many cases what he is calling a movie frame can also be seen as a frame in the sense of this paper as well. Thus an event of a rise in temperature could be seen as a concatenation of two temperature frames, that is, an object of type $\mathit{AmbTemp}^\frown\mathit{AmbTemp}$. We have seen a concatenation type previously in our characterization of a phonology type in (8). That is because phonological events are also to be seen as event strings in Fernando's sense. (11) shows a type of event for a rise in temperature using the temperature frame *AmbTemp* in (2).

(11)
$$\left[\begin{array}{l}\text{e-time}{:}\mathit{TimeInt}\\ \text{start}{:}\left[\begin{array}{l}\text{x}{:}\mathit{Ind}\\ \text{e-time}{=}\text{e-time.start}{:}\mathit{Time}\\ \text{e-location}{:}\mathit{Loc}\\ \text{c}_{\text{temp\_at\_in}}{:}\text{temp\_at\_in(start.e-time, start.e-location, start.x)}\end{array}\right]\\ \text{end}{:}\left[\begin{array}{l}\text{x}{:}\mathit{Ind}\\ \text{e-time}{=}\text{e-time.end}{:}\mathit{Time}\\ \text{e-location}{=}\text{start.e-location}{:}\mathit{Loc}\\ \text{c}_{\text{temp\_at\_in}}{:}\text{temp\_at\_in(end.e-time, end.e-location, end.x)}\end{array}\right]\\ \text{event}{=}\text{start}^\frown\text{end}{:}\mathit{AmbTemp}^\frown\mathit{AmbTemp}\\ \text{c}_{\text{incr}}{:}\text{start.x}{<}\text{end.x}\end{array}\right]$$

Here we make use of *manifest fields* [7] such as

(12) $[\text{e-time}{=}\text{e-time.start}{:}\mathit{Time}]$

which restrict the type in the field to be a singleton type of the unique object represented after the equality sign. Thus (12) is syntactic sugar for

(13) $[\text{e-time}{:}\mathit{Time}_{\text{e-time.start}}]$

This uses a singleton type represented by $\mathit{Time}_{\text{e-time.start}}$. If some object a is of type T ($a : T$) then T_a is a type such that $b : T_a$ iff $b = a$. That is, we

restrict the type to be the type of a unique particular object. It should also be noted that path names such as 'start.e-time' always begin at the root of the record type rather than the most local record type in which they occur. (11) is then the type of events where there is a rise in ambient temperature. An event e of this type will be of type rise(e.start,e.e-time). In fact we will make the stronger requirement that if r:*AmbTemp* and i:*TimeInt* then e:rise(r,i) iff e:(11), e.start=r and e.e-time=i.

6 Word Meaning in Flux

For all (11) is based on a very much simplified version of FrameNet's `Ambient_temperature`, it represents a quite detailed account of the lexical meaning of *rise* in respect of ambient temperature — detailed enough, in fact, to make it inappropriate for *rise* with other kinds of subject arguments. Consider price. The type of a price rising event could be represented by (14).

(14)
$$\left[\begin{array}{ll}
\text{e-time}: \textit{TimeInt} & \\
\text{start}: & \left[\begin{array}{l} \text{x}:\textit{Ind} \\ \text{e-time=e-time.start}:\textit{Time} \\ \text{e-location}:\textit{Loc} \\ \text{commodity}:\textit{Ind} \\ \text{c}_{\text{price_of_at_in}}:\text{price_of_at_in(start.commodity,} \\ \qquad \text{start.e-time, start.e-location, start.x)} \end{array}\right] \\
\text{end}: & \left[\begin{array}{l} \text{x}:\textit{Ind} \\ \text{e-time=e-time.end}:\textit{Time} \\ \text{e-location=start.e-location}:\textit{Loc} \\ \text{commodity=start.commodity}:\textit{Ind} \\ \text{c}_{\text{price_of_at_in}}:\text{price_of_at_in(end.commodity,} \\ \qquad \text{end.e-time, end.e-location, end.x)} \end{array}\right] \\
\text{event=start}^\frown\text{end}:\textit{Price}^\frown\textit{Price} & \\
\text{c}_{\text{incr}}:\text{start.x<end.x} &
\end{array}\right]$$

(14) is similar to (11) but crucially different. A price rising event is, not surprisingly, a string of price frames rather than ambient temperature frames. The type of price frames (*Price*) is given in (15).

(15)
$$\left[\begin{array}{ll}
\text{x} & : \textit{Ind} \\
\text{e-time} & : \textit{Time} \\
\text{e-location} & : \textit{Loc} \\
\text{commodity} & : \textit{Ind} \\
\text{c}_{\text{price_of_at_in}} & : \text{price_of_at_in(commodity, e-time, e-location, x)}
\end{array}\right]$$

If you look up the noun *price* in FrameNet[6] you find that it belongs to the frame `Commerce_scenario` which includes frame elements for goods (corresponding to our 'commodity') and money (corresponding to our 'x'-field). If you compare the

[6] Accessed 8th April, 2010.

FrameNet frames `Ambient_temperature` and `Commerce_scenario`, they may not initially appear to have very much in common. However, extracting out just those frame elements or roles that are relevant for the analysis of the lexical meaning of *rise* shows a degree of correspondence. They are, nevertheless, not the same. Apart from the obvious difference that the predicate in the constraint field that relates the various roles involves temperature in the one and price in the other, price crucially involves the role for commodity since this has to be held constant across the start and end frames. We cannot claim that a price is rising if we check the price of tomatoes in the start frame and the price of oranges in the end frame.

This corresponds to a situation which is familiar to us from work on the Generative Lexicon [20,21] where the arguments to words representing functions influence the precise meaning of those words. For example, *fast* means something different in *fast car* and *fast road*, although, of course, the two meanings are related. There are two important questions that arise when we study this kind of data:

- is it possible to extract a single general meaning of words which covers all the particular meanings of the word in context?
- is it possible to determine once and for all the set of particular contextually determined meanings?

Our suspicion is that the answer to both these questions is "no". It seems that we are able to create new meanings for words based on old meanings to suit the situation that we are currently trying to describe and that there is no obvious requirement that all these meanings be consistent with each other, making it difficult to extract a single general meaning. Here we are following the kind of theory proposed by Larsson and Cooper [14,6]. According to such a theory the traditional meaning question "What is the meaning of expression E?" should be replaced by the following two questions relating to the way in which agents coordinate meaning as they interact with each other in dialogue or, more indirectly, through the writing and reading of text:

the coordination question Given resources R, how can agent A construct a meaning for a particular utterance U of expression E?

the resource update question What effect will this have on A's resources R?

Let us look at a few examples of uses of the verb *rise* which suggest that this is the kind of theory we should be looking at. Consider first that a fairly standard interpretation of *rise* concerns a change in location. (16) is part of the description of a video game.[7]

(16) As they get to deck, they see the Inquisitor, calling out to a Titan in the seas. **The giant Titan rises through the waves**, shrieking at the Inquisitor.

[7] `http://en.wikipedia.org/wiki/Risen\_(video\_game)`, accessed 4th February, 2010.

The type of the rising event described here could be something like (17).

(17)
$$\begin{bmatrix}
\text{e-time}: \textit{TimeInt} \\
\text{start}: \begin{bmatrix} \text{x}:\textit{Ind} \\ \text{e-time}=\text{e-time.start}:\textit{Time} \\ \text{e-location}:\textit{Loc} \\ \text{c}_{\text{at}}:\text{at(start.x,start.e-location,start.e-time)} \end{bmatrix} \\
\text{end}: \begin{bmatrix} \text{x}=\text{start.x}:\textit{Ind} \\ \text{e-time}=\text{e-time.end}:\textit{Time} \\ \text{e-location}:\textit{Loc} \\ \text{c}_{\text{at}}:\text{at(end.x,end.e-location,end.e-time)} \end{bmatrix} \\
\text{event}=\text{start}^\frown\text{end}:\textit{Position}^\frown\textit{Position} \\
\text{c}_{\text{incr}}:\text{height(start.e-location)}<\text{height(end.e-location)}
\end{bmatrix}$$

This relies on a frame type *Position* given in (18).

(18)
$$\begin{bmatrix}
\text{x} & : \textit{Ind} \\
\text{e-time} & : \textit{Time} \\
\text{e-location} & : \textit{Loc} \\
\text{c}_{\text{at}} & : \text{at(x,e-location,e-time)}
\end{bmatrix}$$

(18) is perhaps most closely related to FrameNet's `Locative_relation`. (17) is structurally different from the examples we have seen previously. Here the content of the 'x'-field, the focus of the frame, which in the case of the verb *rise* will correspond to the subject of the sentence, is held constant in the string of frames in the event whereas in the case of rising temperatures and prices it was the focus that changed value. Here it is the height of the location which increases whereas in the previous examples it was important to hold the location constant.[8] This makes it difficult to see how we could give a single type which is general enough to include both varieties and still be specific enough to characterize "the meaning of *rise*". It appears more intuitive and informative to show how the variants relate to each other in the way that we have done.

The second question we had concerned whether there is a fixed set of possible meanings available to speakers of a language or whether speakers create appropriate meanings on the fly based on their previous experience. Consider the examples in (19).

(19) a. Mastercard rises
b. China rises

[8] We have used 'height(start/end.e-location)' in (17) to represent the height of the location since we have chosen to treat *Loc*, the type of spatial location, as a basic type. However, in a more detailed treatment *Loc* should itself be treated as a frame type with fields for three coordinates one of them being height, so we would be able to refer to the height of a location l as l.height.

While speakers of English can get an idea of the content of the examples in (19) when stripped from their context, they can only guess at what the exact content might be. It *feels* like a pretty creative process. Seeing the examples in context as in (20) reveals a lot.[9]

(20)
a. Visa Up on Q1 Beat, Forecast; **Mastercard Rises** in Sympathy
By Tiernan Ray
Shares of Visa (V) and Mastercard (MA) are both climbing in the aftermarket, reversing declines during the regular session, after Visa this afternoon reported fiscal Q1 sales and profit ahead of estimates and forecast 2010 sales growth ahead of estimates, raising enthusiasm for its cousin, Mastercard.
b. The rise of China will undoubtedly be one of the great dramas of the twenty-first century. China's extraordinary economic growth and active diplomacy are already transforming East Asia, and future decades will see even greater increases in Chinese power and influence. But exactly how this drama will play out is an open question. Will China overthrow the existing order or become a part of it? And what, if anything, can the United States do to maintain its position as **China rises**?

It seems like the precise nature of the frames relevant for the interpretation of *rises* in these examples is being extracted from the surrounding text by a technique related to automated techniques of relation extraction in natural language processing.

7 Conclusion

We have suggested that a notion of frame can be of use in an approach to formal semantics dealing with hard empirical questions of lexical semantics and linguistic processing. The important aspect of our analysis is that we have semantic objects corresponding to frames and allow these to be arguments to predicates. We have illustrated this with an old puzzle from formal semantics, the Partee puzzle concerning the rising of temperature. Our solution is very similar in strategy to that originally proposed by Montague. It differs in that we use frames where Montague used individual concepts.

The additional detail of the lexical semantic analysis obtained by using frames comes at a cost, however. It has as a consequence that there is not obviously a single meaning or even a small set of meanings associated with *rise*. Rather *rise* means something slightly different for temperatures and prices, objects rising in

[9] http://blogs.barrons.com/stockstowatchtoday/2010/02/03/visa-up-on-q1-beat-forecast-mastercard-moves-in-sympathy/?mod=rss_BOLBlog, accessed 4th February, 2010; http://www.foreignaffairs.com/articles/63042/g-john-ikenberry/the-rise-of-china-and-the-future-of-the-west, accessed 4th February, 2010.

location, not to mention countries as in *China rises*. This spread of meanings seems to be important if we are to draw the kinds of detailed inferences that speakers of a language are able to draw from these examples.

We have argued that there is no fixed set of meanings but rather that speakers of a language create meanings on the fly for the purposes of interpretation in connection with a given speech (or reading) event. This idea is related to the notion of *meaning potential* discussed for example in [15] and a great deal of other literature. While we have made no precise proposal for how speakers go about creating new situation specific meanings in this paper we believe that the kinds of structured semantic objects (such as frames) that we are proposing in this paper will facilitate an account of this. Our record types comprise a collection of fields (which can be used to correspond to frame elements). New meanings can be constructed from old ones by adding, subtracting or modifying such fields, thus providing possibilities for change that are not so obviously available in traditional possible world semantics based on functions from possible worlds and times to denotations.

Acknowledgements. This research was supported in part by VR project 2009-1569, Semantic analysis of interaction and coordination in dialogue (SAICD) and Swedish Tercentenary Foundation Project P2007/0717, Semantic Coordination in Dialogue (SemCoord). I am grateful to Jonathan Ginzburg and Staffan Larsson for discussion. Previous versions of this material have been presented at the seminar of the Centre for Language Technology in Gothenburg, the linguistics seminar of the Department of Philosophy, Linguistics and Theory of Science at the University of Gothenburg and the Grammar Festival organized by the Department of Swedish at the University of Gothenburg. I am grateful to the audiences on all these occasions for useful discussions and improvements.

References

1. Bos, J., Nissim, M.: Combining Discourse Representation Theory with FrameNet. In: Favretti, R.R. (ed.) Frames, Corpora, and Knowledge Representation, pp. 169–183. Bononia University Press (2008)
2. Cooper, R.: Austinian truth, attitudes and type theory. Research on Language and Computation 3, 333–362 (2005)
3. Cooper, R.: Records and record types in semantic theory. Journal of Logic and Computation 15(2), 99–112 (2005)
4. Cooper, R.: Type theory with records and unification-based grammar. In: Hamm, F., Kepser, S. (eds.) Logics for Linguistic Structures, pp. 9–34. Mouton de Gruyter, Berlin (2008)
5. Cooper, R.: Type theory and semantics in flux. In: Kempson, R., Asher, N., Fernando, T. (eds.) Handbook of the Philosophy of Science. Philosophy of Linguistics, vol. 14. Elsevier BV, Amsterdam (forthcoming), General editors Gabbay, D.M., Thagard, P., Woods, J.
6. Cooper, R., Larsson, S.: Compositional and ontological semantics in learning from corrective feedback and explicit definition. In: Edlund, J., Gustafson, J., Hjalmarsson, A., Skantze, G. (eds.) Proceedings of DiaHolmia: 2009 Workshop on the Semantics and Pragmatics of Dialogue, Department of Speech, Music and Hearing, KTH, pp. 59–66 (2009)

7. Coquand, T., Pollack, R., Takeyama, M.: A logical framework with dependently typed records. Fundamenta Informaticae XX, 1–22 (2004)
8. Fernando, T.: A finite-state approach to events in natural language semantics. Journal of Logic and Computation 14(1), 79–92 (2004)
9. Fernando, T.: Situations as strings. Electronic Notes in Theoretical Computer Science 165, 23–36 (2006)
10. Fernando, T.: Finite-state descriptions for temporal semantics. In: Bunt, H., Muskens, R. (eds.) Computing Meaning. Studies in Linguistics and Philosophy, vol. 3, vol. 83, pp. 347–368. Springer, Heidelberg (2008)
11. Fernando, T.: Situations in LTL as strings. Information and Computation 207(10), 980–999 (2009)
12. Fillmore, C.J.: Frame semantics. In: Linguistics in the Morning Calm, pp. 111–137. Hanshin Publishing Co., Seoul (1982)
13. Ginzburg, J.: The Interactive Stance: Meaning for Conversation. Oxford University Press, Oxford (forthcoming)
14. Larsson, S., Cooper, R.: Towards a formal view of corrective feedback. In: Alishahi, A., Poibeau, T., Villavicencio, A. (eds.) Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition. EACL, pp. 1–9 (2009)
15. Linell, P.: Rethinking Language, Mind, and World Dialogically: Interactional and contextual theories of human sense-making. In: Advances in Cultural Psychology: Constructing Human Development, Information Age Publishing, Inc., Charlotte (2009)
16. Montague, R.: The Proper Treatment of Quantification in Ordinary English. In: Hintikka, J., Moravcsik, J., Suppes, P. (eds.) Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics, pp. 247–270. D. Reidel Publishing Company, Dordrecht (1973)
17. Montague, R.: Formal Philosophy: Selected Papers of Richard Montague. Yale University Press, New Haven (1974), Introduction by R.H. Thomason (ed.)
18. Prior, A.N.: Time and modality. Oxford University Press, Oxford (1957)
19. Prior, A.N.: Past, present and future. Oxford University Press, Oxford (1967)
20. Pustejovsky, J.: The Generative Lexicon. MIT Press, Cambridge (1995)
21. Pustejovsky, J.: Type theory and lexical decomposition. Journal of Cognitive Science 6, 39–76 (2006)
22. Reichenbach, H.: Elements of Symbolic Logic. University of California Press, Berkeley (1947)

Clustering E-Mails for the Swedish Social Insurance Agency – What Part of the E-Mail Thread Gives the Best Quality?

Hercules Dalianis[1], Magnus Rosell[1,2], and Eriks Sneiders[1]

[1] Department of Computer and Systems Science,
(DSV) Stockholm University
Forum 100, 164 40 Kista, Sweden
[2] KTH CSC, 100 44 Stockholm, Sweden
hercules@dsv.su.se,rosell@csc.kth.se,eriks@dsv.su.se

Abstract. We need to analyse a large number of e-mails sent by the citizens to the customer services department of a governmental organisation based in Sweden. To carry out this analysis we clustered a large number of e-mails with the aim of automatic e-mail answering. One issue that came up was whether we should use the whole e-mail including the thread or just the original query for the clustering. In this paper we describe this investigation. Our results show that only the query and the answering part should be used, but not necessarily the whole e-mail thread. The results clearly show that the original question contains more useful information than only the answer, although a combination is even better. Using the full e-mail thread does not downgrade the result.

Keywords: E-government, query answering, e-mail threads, Swedish, clustering.

1 Introduction

In Sweden the public authorities have been in the lead to implement E-government. This includes communication with the citizens through various electronic channels. One such channel is to put important information on their web sites. Citizens often do not find the information they are seeking, however, and initiate communication in one of several ways, such as telephone calls, e-mails, chat lines, etc.

The Swedish Social Insurance Agency[1] (SSIA) receives more than 10 000 e-mails from citizens each week. These are answered manually by handling officers. Many of the e-mails from the public are very similar. Therefore a lot would be gained if these re-occurring questions could be answered automatically or semi-automatically. To accomplish this, first the common questions must be identified.

The e-mails are either sent directly to an available address or via a web form on the agency's web site. When a citizen uses the web form he/she also has to assign

[1] www.forsakringskassan.se

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 115–120, 2010.

a category to it, such as parental benefit (föräldrapenning), housing allowance (bostadsbidrag), superannuation (pension), sickness benefit (sjukpenning), etc.

Although these broad categories help to assign the e-mails to the right handling officer they do not help to identify the common questions. To find groups of common questions we apply text clustering.

Our ultimate goal is to help the handling officers to use clustering as a tool to facilitate more efficient and up-to-date answers. Clustering could be used to get an overview of the trends in the questions and to identify common questions that could be answered using a standard answer. In the long run such questions could also be answered automatically, [1].

In the present work we investigate clustering of the e-mails without involving the handling officers. We study the effect of using different parts of the e-mail threads in order to achieve the best clustering quality.

2 Previous Research

An e-mail consists of a header (including sender and receiver addresses, subject matter, etc) and body text. The body text may also be divided into several zones of different kinds of content, such as sender zones *(author, greeting, signoff)*, quoted conversation zones *(reply, forward)*, and boilerplate zones *(signatures, advertising, disclaimer, attachment)* [2].

Previous work on clustering of e-mails has discussed the inclusion of different parts of the e-mails, but has not tried different parts of the body. In [3] using a combination of the header and body gives better results than using only the body. In [4] the authors let the user weight the importance of the parts *(to, cc, from, subject, date, body)*.

Whereas previous research was aimed at personal inboxes, we study e-mails sent to a whole organisation.

3 Text Sets and Preprocessing

We received about 9 000 e-mails from the SSIA. Around 4 000 of these were either sent directly (without the use of the web form) or assigned a miscellaneous category "other questions" (övriga frågor) in the web form. As we could not use these for our evaluation we removed them, producing a set of almost 5 000 e-mails that were categorised.

All e-mails were also de-identified because of their sensitive nature. The de-identification of the e-mails was carried out by SSIA before the e-mails were handed over to our research group. The de-identification program was developed and evaluated by our research group. For first names we obtained an F-score of 0.82 and a recall of 0.73 respectively and for last names an F-score of 0.85 and a recall of 0.77 respectively. For social security numbers, phone numbers, e-mail addresses, web addresses, street addresses and postal codes we obtained an F-score of 0.93 and a recall of 0.92.

3.1 Extracting Parts of the E-Mail Thread

The e-mails we obtained were actually complete e-mail threads as they had developed up until the moment they were extracted at the SSIA. The number of items in a thread varied from one to 40 although 96.2 percent of all threads where up to four components long.

The principle of separating thread components was empirically obtained by working on a large number of e-mails. The system iteratively cuts off the top message. It first looks for several successive lines that start with ">".

If these are found, then everything above these lines is the top message. Otherwise the system looks for a typical message separator line, such as "abc@doc.com wrote:", "Original message:", etc in several languages (Swedish, English, Norwegian) with a certain level of wording freedom. If this does not help, it looks for an array of lines that start with "From:", "To:", "Date:", "Subject:" in different languages. This method is based on heuristics but works comparably well.

For our clustering experiments we created four sets of texts: *Question* – a set containing only the first question in each thread, *Answer* – the first answer to each question, *Question and Answer* – the first question and the first answer, and *Thread* – the whole e-mail thread.

3.2 Lemmatisation and Filtering

Using a few simple rules we removed the e-mail headers and characters indicating quotation/citation of previous messages in the thread. We were not allowed to use the headers due to the sensitive nature of these e-mails.

The results for each of the different sets were tokenised and lemmatised using the Swedish grammar checking program Granska [5]. The resulting texts still contained a lot of non-word character sequences, coming from signatures, advertisements, disclaimers, etc. To try to remove them we have used several simple methods. We removed words shorter than three characters and longer than 20, since this only removes a few interesting words and captures some of the non-words. Further, we removed all words only appearing in only one e-mail, (see appendix in [6]), since they did not contribute to the similarity between e-mails. We also used a common stoplist of Swedish words.

3.3 Statistics for the Preprocessed Text Sets

Table 1 gives some statistics for the extracted and preprocessed text sets: the number of texts and lemmas, as well as the average number of different lemmas per text and the average number of texts in which each lemma occurs.

4 Clustering

For each text set we constructed an ordinary term-document-matrix with tf*idf-weights. We defined similarity between texts as the cosine measure.

We have used the K-Means algorithm, (see for instance [7]), as it is simple, fast, and therefore suitable for interactive exploration. In the end we want the handling officers to use clustering as a tool to obtain an overview of the trends in the questions and to indentify common questions, this in an interactive manner as described in [8].

5 Evaluation

Since internal clustering quality measures are based on the representation we can not use them to compare results based on different representations, i.e. our text sets. External quality measures compare the clustering with a categorisation. We have the categorisation made by the citizens. It may not be ideal, but at least it groups questions with similar content. We want clusterings to compare well with this categorisation, although we do not expect them to be very similar. We prefer a clustering to be more similar rather than less similar, however.

There are many external quality measures. We prefer information theory based measures as these take the whole distribution of texts over categories and clusters into account. For this reason we use the Normalised Mutual Information (NMI) between the clustering and the categorisation, see [9].

Table 1. Clustering results for four different text sets (based on the original question only, the first answer only, both first question and answer, and the full e-mail thread). The first four measures describe the text sets after preprocessing. The last measure is the average clustering result in NMI (Normalised Mutual Information) of 20 K-Means clusterings to nine clusters compared with the categorisation. Standard deviations are shown in parenthesis.

	Text Set			
Measure	Question	Answer	Question and Answer	Thread
Number of Texts	4 652	4 681	4 839	4 841
Number of Lemmas	2 929	2 055	3 956	4 398
Lemmas/Text	12.2	9.2	19.5	23.2
Texts/Lemma	19.3	21.0	23.9	25.5
NMI	0.28 (0.03)	0.14 (0.02)	0.40 (0.03)	0.38 (0.04)

6 Experiments and Discussion

In Table 1 we report average results in NMI for nine to 20 clusterings of the different text sets, with the standard deviation shown in parenthesis. In order for two results to be considered different they, as a rule of thumb, they need not overlap with their standard deviations.

We choose nine clusters as the categorisation has nine categories. The tendencies we describe are similar for other numbers of clusters.

The result clearly shows that the textual information in the question (*Question*) is better than what is in the answer (*Answer*). The result gets even better,

however, if we also include the answer (*Question and Answer*). The result for the entire e-mail thread (*Thread*) is the same as for *Question and Answer*. As shown in Table 1 the Normalised Mutual Information (NMI) for the query and answering part is 0.12 units higher than for only the query alone.

It is not surprising that the result is better for the set of questions than for the set of answers as the categories are chosen by the citizens who also formulated the questions. The answers are often shorter than the questions (see the statistics in Table 1), use a more formal language, and do not necessarily include the same terms as their corresponding question. This makes the answers harder to group. Combined with the question, however, the answer does give more information (than using only the question) for the clustering algorithm to work with as similar questions tend to be answered in similar ways.

By the same reasoning, the result when using the entire thread is used should be even better. The questions that require more responses (follow-up questions with answers), however, are probably more complicated and therefore harder to group. The categorisation of the first question might not even be suitable for the entire thread as it may well include new questions regarding other matters.

As the full thread (*Thread*) contains most information and it performs equally well with the questions and answers (*Question and Answer*) we will use it in our further work.

7 Conclusions and Future Work

We have compared clusterings of e-mails sent to the SSIA based on different parts of the e-mail thread/texts. The results clearly show that the original question contains more useful information than only the answer, although a combination is even better. Using the full e-mail thread does not downgrade the result.

We plan to involve the handling officers in our next investigation. We will let them explore clusterings of the e-mails and interview them to learn whether an approach like this is actually useful and if it can provide insights, help to find common questions and formulate standard answers.

Acknowledgements. We would like to thank Anne-Lie Karlsson at Försäkringskassan, SSIA, for her devoted support of the IMAIL research group. We would also like to thank VINNOVA (The Swedish Governmental Agency for Innovation Systems) for the funding of the IMAIL-project.

References

1. Knutsson, O., Pargman, T., Dalianis, H., Rosell, M., Sneiders, E.: Increasing the efficiency and quality of e-mail communication in e-Governmnent using language technology. In: Proc. of IFIP e-Government Conference 2010 (EGOV 2010), Lausanne, Switzerland, August 29-September 2 (2010) (to be published)
2. Lampert, A., Dale, R., Paris, C.: Segmenting email message text into zones. In: Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009 (2009)

3. Huang, Y., Govindaraju, D., Mitchell, T.M., de Carvalho, V.R., Cohen, W.W.: Inferring ongoing activities of workstation users by clustering email. In: CEAS – Conference on Email and Anti-Spam (2004)
4. Schuff, D., Turetken, O., D'Arcy, J.: A multi-attribute, multi-weight clustering approach to managing "e-mail overload". Decision Support Systems 42, 1350–1365 (2006)
5. Domeij, R., Knutsson, O., Carlberger, J., Kann, V.: Granska – an efficient hybrid system for Swedish grammar checking. In: Proc. 12th Nordic Conf. on Comp. Ling. – NODALIDA 1999 (1999)
6. Rosell, M.: Text Clustering Exploration – Swedish Text Representation and Clustering Results Unraveled. PhD thesis, School of Computer Science and Communication, Royal Institute of Technology, Stockholm, Sweden (2009)
7. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
8. Cutting, D.R., Pedersen, J.O., Karger, D., Tukey, J.W.: Scatter/Gather: A cluster-based approach to browsing large document collections. In: Proc. 15th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (1992)
9. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3, 583–617 (2003)

OpenMaTrEx: A Free/Open-Source Marker-Driven Example-Based Machine Translation System

Sandipan Dandapat[1], Mikel L. Forcada[1,2], Declan Groves[1,3], Sergio Penkale[1], John Tinsley[1], and Andy Way[1]

[1] Centre for Next Generation Localisation, School of Computing, Dublin City University, Glasnevin, Dublin 9, Ireland
[2] Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, E-03071 Alacant, Spain
[3] Traslán Teoranta, 31 Mill Grove, Killincarrig, Greystones, Co. Wicklow, Ireland
{sdandapat,mforcada,dgroves,spenkale,jtinsley,away}@computing.dcu.ie

Abstract. We describe OpenMaTrEx, a free/open-source example-based machine translation (EBMT) system based on the marker hypothesis, comprising a marker-driven chunker, a collection of chunk aligners, and two engines: one based on a simple proof-of-concept monotone EBMT recombinator and a Moses-based statistical decoder. OpenMaTrEx is a free/open-source release of the basic components of MaTrEx, the Dublin City University machine translation system.

Keywords: example-based machine translation, corpus-based machine translation, free/open-source software.

1 Introduction

We describe OpenMaTrEx, a free/open-source (FOS) example-based machine translation (EBMT) system based on the marker hypothesis [1]. It comprises a marker-driven chunker, a collection of chunk aligners, and two engines: one based on the simple proof-of-concept monotone recombinator (previously released as *Marclator*[1]) and a Moses-based decoder [2]. OpenMaTrEx is a FOS version of the basic components of MaTrEx, the Dublin City University machine translation (MT) system [3,4]. Most of the code in OpenMaTrEx is written in Java, although there are many important tasks that are performed in a variety of scripting languages. A preliminary version, 0.71, has been released for download from `http://www.openmatrex.org` on 2nd June 2010, under a FOS licence.[2]

The architecture of OpenMaTrEx is the same as that of a baseline MaTrEx system [3,4]; as MaTrEx, it can wrap around the Moses statistical MT

[1] `http://www.openmatrex.org/marclator/`
[2] GNU GPL version 3, `http://www.gnu.org/licenses/gpl.html`

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 121–126, 2010.

decoder, using a hybrid translation table containing marker-based chunks as well as statistically extracted *phrase*[3] pairs.

OpenMaTrEx has been released as a FOS package so that MaTrEx components which have successfully been used [5,6,7] may be combined with components from other FOS machine translation (FOSMT) toolkits such as Cunei[4] [8], Apertium[5] [9], etc.[6] Indeed, using components released in OpenMaTrEx, researchers have previously: used statistical models to rerank the results of recombination [10]; used aligned, marker-based chunks in an alternative decoder which uses a memory-based classifier [11]; combined the marker-based chunkers with rule-based components [12], and used the chunker to filter out Moses phrases for linguistic motivations [13].

The rest of the paper is organized as follows. Section 2 describes the principles of training and translation in OpenMaTrEx; section 3 describes the EBMT-specific components in OpenMaTrEx; section 4 describes its software requirements and briefly explains how to install and run the available components. A sample experiment performed on a standard task with OpenMaTrEx is described in section 5 and results are compared to those obtained with a a standard statistical machine translation (SMT) system. Concluding remarks are made in section 6.

2 OpenMaTrEx: Training and Translation

Training with OpenMaTrEx may be performed in two different modes. In MaTrEx *mode:*

1. Each example sentence in the sentence-aligned source text and its counterpart in the target training text are divided in subsentential segments using a marker-based *chunker*. Chunks may optionally be tagged according to their initial marker word (to further guide the alignment process).
2. A complete Moses–GIZA++[7] training run is performed up to Moses step 5 (phrase extraction). Moses is used to learn a maximum-likelihood lexical translation table and to extract phrase-pair tables.
3. The subsentential chunks are aligned using one of the *aligners* provided (using, among other information, probabilities generated by GIZA++).
4. Aligned chunk pairs from step 3 are *merged* with the phrase pairs generated by Moses in step 2 (more details in section 3).
5. From then on, training proceeds as a regular Moses job after Moses step 6. MERT [14] may be used on a development set for tuning.

In *Marclator* mode (see below), the last two steps are not necessary and Moses is only run up to step 4.

[3] In statistical MT, the term *phrase* is stretched to refer to any contiguous sequence of words.

[4] http://www.cunei.org

[5] http://www.apertium.org

[6] For a longer list of FOSMT systems, visit http://fosmt.info

[7] http://www.fjoch.com/GIZA++.html

Translation may be performed, as training, in two ways:

- *Marclator mode* uses a monotone ("*naïve*") decoder (released as part of Marclator): each source sentence is run through the marker-based chunker; the most probable translations for each chunk are retrieved, along with their weights; if no chunk translations are found, the decoder backs off to the most likely translations for words (as aligned by GIZA++) and concatenates them in the same order, and when no translation is found, leaves any unfound source words untranslated. This decoder has obvious limitations, but it is fast and likely to be of most use in the case of closely related language pairs.
- MaTrEx *mode*, however, is the usual way to use OpenMaTrEx; that is, the Moses decoder is run on a merged phrase table, as in MaTrEx [3,4].

3 EBMT-Specific Components

Chunker. The main chunker in OpenMaTrEx is based on the *marker hypothesis* [1] which states that the syntax of a language is marked at the surface level by a set of marker (closed-category) words or morphemes. The chunker in OpenMaTrEx deals with left-marking languages: a chunk starts at a marker word, and must contain at least one non-marker word. Punctuation is also used to delimit chunks. Version 0.71 provides marker files for Catalan, Czech, English, Portuguese, Spanish, Irish, French and Italian. Marker files specify one marker word or punctuation in each line: its surface form, its category and (optionally) its subcategory. A typical marker word file contains a few hundred entries.

Chunk aligners. There are a number of different chunk aligners available in OpenMaTrEx. The default aligner aligns chunks using a regular Levenshtein edit distance with a combination of costs specified in a configuration file, optionally allowing *jumps* or block movements [3]. The default combination uses two costs: a *probability cost* based on word translation probabilities as calculated by using GIZA++ and Moses (see training step 2 in section 2), and a *cognate cost* based on a combination of the Levenshtein distance, the longest common subsequence ratio and the Dice coefficient. As in [3], equal weights are used as a default for all component costs specified.

Translation table merging. To run the system in MaTrEx *mode*, marker-based chunk pairs are merged with phrase pairs from alternative resources (here, Moses phrases). Firstly, each chunk pair is assigned a word alignment based on the refined GIZA++ alignments, for example "`please show me ||| por favor muéstreme ||| 0-0 0-1 1-2 2-2`". In cases where there is no word alignment for a particular chunk pair according to GIZA++, the chunk pair is discarded. Using these word alignments, we additionally extract a phrase orientation-based lexicalised reordering model *à la* Moses [15]. Finally, we may also limit the maximum length of chunks pairs that will be used. The resulting chunk pairs are in the same format as those phrase pairs extracted by Moses. The next step is to combine the chunk pairs with Moses phrase pairs. In order to do this, the two sets of chunk/phrase pairs are merged into a single file. Moses training is

then carried out from step 6 (*scoring*) which calculates the required scores for all feature functions, including the reordering model, based on the combined counts. A binary feature distinguishing EBMT chunks from SMT chunks may be added for subsequent MERT optimization as was done in [16].

4 Technical Details

Required software. OpenMaTrEx requires the installation of the following software: GIZA++, Moses, IRSTLM [17], and a set of auxiliary scripts for corpus preprocessing[8] and evaluation (mteval).[9] Refer to the `INSTALL` file that comes with the distribution for details.

Installing OpenMaTrEx itself. OpenMaTrEx may easily be built simply by invoking `ant` or an equivalent tool on the `build.xml` provided. The resulting `OpenMaTrEx.jar` contains all the relevant classes, some of which will be invoked using a shell, `OpenMaTrEx` (see below).

Running. A shell (`OpenMaTrEx`) has options to initialise the training, development, and testing sets, to call the chunker and the aligner, to train a target language model with IRSTLM, to run GIZA++ and Moses training jobs, to merge marker-based chunk pairs with Moses phrase pairs, to run MERT optimization jobs, and to execute the decoders. Future versions will contain higher-level ready-made options for the most common training and translation jobs. For detailed instructions on how to perform complete training and translation jobs in both MaTrEx and *Marclator* mode, see the `README` file. Test files will be provided in the `examples` directory of the OpenMaTrEx package.

5 A Sample Experiment

To show how OpenMaTrEx can be used to improve baseline SMT results, we report on a simple experiment using 200,000 randomly selected sentences from the Spanish–English Europarl corpus provided for the Third Workshop on SMT (WMT08): testing was performed on the 2,000-sentence test set provided by WMT08. The experimental conditions are the same as those reported in [16]. Table 1 shows results for (i) a baseline Moses job, (ii) a job in which marker-based chunk pairs were transformed into Moses translation table pairs as described in section 3 and simply appended to the Moses phrase pairs, and (iii) a third job in which an extra feature (having the value 1 for marker-based chunk pairs and 0 for Moses-extracted phrase pairs) is added to the usual five features in all phrase pairs before MERT tuning. The table shows BLEU and NIST scores as well as the fraction of phrase pairs used during translation that were extracted by the marker-based chunker and aligner. Clearly, using the feature-informed phrase table merging improves the BLEU (with 93% statistical significance [18]) and NIST scores (76% confidence), while simple merging does not seem to help. These improvements correlate

[8] `http://homepages.inf.ed.ac.uk/jschroe1/how-to/scripts.tgz`
[9] We currently use version 11b from `ftp://jaguar.ncsl.nist.gov/mt/resources/`

Table 1. A sample experiment using 200,000 randomly-selected sentences from the Spanish-English fraction of Europarl, as provided for the Third Workshop on SMT (WMT08). Testing was performed on the 2,000-sentence test set provided by WMT08.

System	BLEU	NIST	EBMT pairs
Baseline Moses	30.59%	7.5171	27.60%
Simple merging	30.42%	7.5156	29.53%
Feature-based merging	30.75%	7.5269	33.55%

nicely with the number of marker-based chunks actually used during translation. It would be interesting to pursue a more detailed study of the actual differences in the translations produced when using more linguistically-motivated chunk pairs.

6 Concluding Remarks and Future Work

We have presented OpenMaTrEx, a FOS EBMT system including a marker-driven chunker (with marker word files for a few languages), chunk aligners, a simple monotone recombinator, and a wrapper around Moses so that it can be used as a decoder for a merged translation table containing Moses phrases and marker-based chunk pairs. OpenMaTrEx releases the basic components of MaTrEx, the Dublin City University machine translation system under a FOS license, to make them available to researchers and developers of MT systems.

As for future work, version 1.0 will contain, among other improvements, a better set of marker files, improved installing and running procedures with extensive training and testing options, and improved documentation; further versions are expected to free/open-source additional MaTrEx components.

Acknowledgements. The original MaTrEx code on which OpenMaTrEx is based was developed among others by S. Armstrong, Y. Graham, N. Gough, D. Groves, H. Hassan, Y. Ma, B. Mellebeek, N. Stroppa, J. Tinsley, and A. Way. We specially thank Y. Graham and Y. Ma for their advice. P. Pecina helped with Czech markers and Jim O'Regan with Irish markers. M.L. Forcada's sabbatical stay at Dublin City University is supported by Science Foundation Ireland (SFI) through ETS Walton Award 07/W.1/I1802 and by the Universitat d'Alacant (Spain). Support from SFI through grant 07/CE/I1142 is acknowledged.

References

1. Green, T.: The necessity of syntax markers. two experiments with artificial languages. Journal of Verbal Learning and Behavior 18, 481–496 (1979)
2. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Ann. Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, pp. 177–180 (June 2007)

3. Stroppa, N., Way, A.: MaTrEx: DCU machine translation system for IWSLT 2006. In: Proceedings of IWSLT 2006, pp. 31–36 (2006)
4. Stroppa, N., Groves, D., Way, A., Sarasola, K.: Example-based machine translation of the Basque language. In: Proc. of AMTA 2006, Cambridge, MA, USA, pp. 232–241 (2006)
5. Groves, D., Way, A.: Hybrid example-based SMT: the best of both worlds? In: ACL-2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond, vol. 100, pp. 183–190 (2005)
6. Hassan, H., Ma, Y., Way, A., Dublin, I.: MaTrEx: the DCU machine translation system for IWSLT 2007. In: Proc. of IWSLT 2007, Trento, Italy, pp. 69–75 (2007)
7. Tinsley, J., Ma, Y., Ozdowska, S., Way, A.: MaTrEx: the DCU MT system for WMT 2008. In: Proc. of the Third Workshop on Statistical Machine Translation, Waikiki, HI, pp. 171–174 (2008)
8. Phillips, A.B., Brown, R.D.: Cunei machine translation platform: System description. In: Proc. of the 3rd Workshop on Example-Based Machine Translation, Dublin, Ireland, pp. 29–36 (November 2009)
9. Tyers, F.M., Forcada, M.L., Ramírez-Sánchez, G.: The Apertium machine translation platform: Five years on. In: Proc. of the First Intl. Workshop on Free/Open-Source Rule-Based Machine Translation, Alacant, Spain, November 2009, pp. 3–10 (2009)
10. Groves, D., Way, A.: Hybridity in MT: Experiments on the Europarl corpus. In: Proc. of the 11th Ann. Conf. of the European Association for Machine Translation (EAMT-2006), Oslo, Norway, pp. 115–124 (2006)
11. van den Bosch, A., Stroppa, N., Way, A.: A memory-based classification approach to marker-based EBMT. In: Proc. of the METIS-II Workshop on New Approaches to Machine Translation, Leuven, Belgium, pp. 63–72 (2007)
12. Sánchez-Martínez, F., Forcada, M.L., Way, A.: Hybrid rule-based – example-based MT: Feeding Apertium with sub-sentential translation units. In: Proc. of the 3rd Workshop on Example-Based Machine Translation, Dublin, Ireland, pp. 11–18 (November 2009)
13. Sánchez-Martínez, F., Way, A.: Marker-based filtering of bilingual phrase pairs for SMT. In: Proc. of EAMT 2009, the 13th Ann. Meeting of the European Association for Machine Translation, Barcelona, Spain, pp. 144–151 (2009)
14. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proc. 41st Ann. Meeting of the Association for Computational Linguistics, Sapporo, Japan, vol. 1, pp. 160–167 (2003)
15. Koehn, P., Axelrod, A., Mayne, A.B., Callison-Burch, C., Osborne, M., Talbot, D.: Edinburgh system description for the 2005 IWSLT speech translation evaluation. In: Proc. of IWSLT 2005, Pittsburgh, PA (2005)
16. Srivastava, A., Penkale, S., Groves, D., Tinsley, J.: Evaluating syntax-driven approaches to phrase extraction for MT. In: Proc. of the 3rd Workshop on Example-Based Machine Translation, Dublin, Ireland, pp. 19–28 (November 2009)
17. Federico, M., Cettolo, M.: Efficient handling of n-gram language models for statistical machine translation. In: Proc. of the 2nd Workshop on Statistical Machine Translation, Prague, Czech Rep., pp. 88–95 (2007)
18. Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proceedings of EMNLP, vol. 4, pp. 388–395 (2004)

Head Finders Inspection: An Unsupervised Optimization Approach

Martín A. Domínguez[1] and Gabriel Infante-Lopez[1,2]

[1] Grupo de Procesamiento de Lenguaje Natural
Universidad Nacional de Córdoba - Argentina
{mdoming,gabriel}@famaf.unc.edu.ar
[2] Consejo Nacional de Investigaciones Científicas y Técnicas

Abstract. Head finder algorithms are used by supervised parsers during their training phase to transform phrase structure trees into dependency ones. For the same phrase structure tree, different head finders produce different dependency trees. Head finders usually have been inspired on linguistic bases and they have been used by parsers as such. In this paper, we present an optimization set-up that tries to produce a head finder algorithm that is optimal for parsing. We also present a series of experiments with random head finders. We conclude that, although we obtain some statistically significant improvements using the optimal head finder, the experiments with random head finders show that random changes in head finder algorithms do not impact dramatically the performance of parsers.

Keywords: Syntactic Parsing, Head Finders, Genetic Algorithms, Bilexical Grammar.

1 Introduction

Head-finder algorithms are used by supervised syntactic parsers to transform phrase structure trees into dependency ones. The transformation is carried out by selecting a word as the head in every constituent. Head-finder algorithms are based on a set of head-finder rules which provides instructions on how to find the head for every type of constituent. For every internal node of a tree, the head-finder rules specify which children of the node contains the head word. The first set of head-rules, based on linguistic principles, was introduced in [1] and it is used by many state-of-the-art statistical parsers, like [2,3,4,5] with only minimal changes.

The standard set of head-finder rules was handcrafted and, consequently, not optimized for parsing; therefore, there might exist different sets of head-finder rules that can improve parsing performance. In this paper we investigate their role in parsing and we experiment with two different state of the art parsers. We present an optimization algorithm that improves the standard set of head finders, one rule at the time, with the goal of finding an optimal set of rules. Even though our optimization algorithm produces statistically significant improvements, they hardly obtain a better performance. In order to better understand why our optimization algorithm cannot produce bigger improvements, we test the stability of the search space. We test this by generating different head finders: we generate head finders that always select the right most and left

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 127–137, 2010.

most subtrees as the trees containing the headword. We also generate 137 random sets of rules, and we test head finders that are not consistent, that is, head finders whose set of rules change during the same training session.

Our optimization procedure aims at finding the best possible set of rules that improves parsing performance. Our procedure is defined as an optimization problem, and as such, it defines the quality measure that it has to optimize, its search space, and the strategy it should follow to find the optimal set of rules among all possible solutions. The *search space* is the possible sets of rules; our procedure optimizes one rule in the set at a time. A new set of rules is then created by replacing an original rule in the standard set with its optimized rule. The *quality measure* for a rule set is computed in a serie of steps. First, the training material is transformed from phrase structure trees into dependency ones using the rule to be evaluated; second, a bilexical grammar [6] is induced from the dependency tree bank, and finally, the quality of the bilexical grammar is evaluated. The quality of the grammar is given by the perplexity (PP) and missed samples (MS) found in the automata of the grammar as explained in Section 3.3. Finally, the *strategy* for traversing the search space is implemented by means of Genetic Algorithms. Once we obtain an optimized set of rules, we proceed to evaluate its impact in two parsers, Collins's parser [5] by means of Bikel's implementation [4], and the Stanford parser [7].

These two parsers have their source code available and their head finder algorithms are rather easy to modify. We considered experimenting also with the Maltparser [8] but its performance is hard to evaluate when its head finder is modified. Our experiments show that the parsing performance of the two parsers is insensitive to variations in head finders. They also show that among all possible head-finders, our optimization procedure is capable of finding improvements. Our experiments also show that in the presence of inconsistent head finder rules, parsers performance drops 1.6% and 0.9% for Bikel's and for Stanford respectively. Our experimental results with random head finders show that modifications in the rule for `VP` produced the biggest impact in the performance of the two parsers. More interestingly, our experiments show that inconsistent head finders are more stable than random deterministic head finder. We argue that this is the case because the variance on the structures the later produce is considerably bigger with respect to the former. Our experiments also show that Stanford parser performance is more stable with respect to variations in the head finder rules than Bikel's.

All in all, our experiments show that, even though it is possible to find some new set of rules that improves parsers performance, head finding algorithms do not have a decisive impact on the performance of these two state-of-the-art syntactic parsers; this also indicates that the reason for their performance lies beyond the procedure that is used to obtain dependencies.

The rest of the paper is organized as follows. Section 2 explains head finding algorithms. Section 3 presents the quality measure used in our optimization algorithm, while Section 4 discusses the search space and the strategy to traverse it. Section 5 presents how random rules are generated, Section 6 presents the results of our experiments and Section 7 introduces related work. Finally, Section 8 concludes the paper.

2 Head Finding Algorithms

For each internal node of a phrase structure tree, the head finder (HF), determines which of its subtrees contains the head word of the constituent. The procedure of transforming a phrase structure into a dependency one starts in the root of the tree and moves downwards up to the tree preterminals. The HF has as a parameter a set of head finder rules R. R contains one rule for each possible grammatical category. Formally, let R be $\{r_{gc_1},\ ...,\ r_{gc_k}\}$, where r_{gc_i} is the head-finder rule associated with the grammatical category gc_i; the set $\{gc_1,\ ...,\ gc_k\}$ is the set of all grammatical category tags like `S`, `VP`, `ADJP`, `ADVP`, `SBAR`, for the Peen Tree-Bank [9] (PTB).

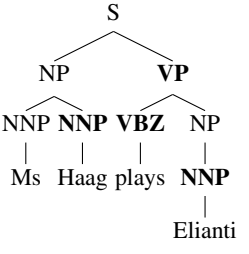


Fig. 1. Sentence 2 from Section 2 of the PTB. The nodes where the head of each constituent is searched for is marked in boldface.

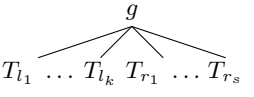


Fig. 2. A simple phrase structure tree

A head finding rule r_{gc} is a vector of pairs $((d_1, t_1), \ldots, (d_{k_n}, t_{k_n}))$, where t_i is a non-terminal and $d_m \in \{left,\ right\}$ is a *search direction*. We also use the notation of *direction vector* to refer to the vector $(d_1, \ldots, d_{k_i})$ which is the projection of the first component of the head-finding rule vector. Similarly, the *tags vector* $(t_1, \ldots, t_{k_i})$ is the projection of the second component. We refer to a head-finder rule as one vector of pairs, or as a pair of vectors. For example, `((l TO)(l VBD)(l VBN)(l MD)(l VBZ)(l VB)(l VBG)(l VBP)(l VP)(l ADJP)(l NN)(l NNS)(l NP))` is the rule that is associated with tag `VP` in the standard set of head-finder rules.

It is important to highlight that our definition of head-finder rule is a simplification of the standard head-finder rule. In the standard definition of rules for tags `NP` and `NX`, sets of non-terminals are used instead of simple non-terminals. Our definition excludes this situation because, otherwise, the size of the search space makes the optimization procedure unfeasible.

In order to show how the head finder algorithm works, we introduce a few auxiliary functions. $root(T)$ returns the root node of the phrase structure tree T; $children(T, g)$ returns the list of children of node g in T and, $subtreeList(T)$ returns the list of subtrees of T ordered from left to right. For example, if T is the tree in Figure 2 then $root(T) = g$, $children(T, g) = [root(T_{i_1}), \ldots, root(T_{l_k}), root(T_{r_1}), \ldots, root(T_{r_s})]$ and $subtreeList(T) = [T_{l_1}, \ldots, T_{l_k}, T_{r_1}, \ldots, T_{r_s}]$. Using this definition, we formally define in Figure 3 the algorithm HF_R that transforms a phrase structure tree into a dependency one, where R is a set of head finder rules.

1. **Let** $r_{gc_i} = ((d_1, t_1), .., (d_{k_i}, t_{k_i}))$ **be** a rule from R defined for tag $root(T) = g_r$
2. **Let** s **be** the size of list $children(T, g_r)$
3. **foreach** $(d_l, t_l) \in r_{gc_i}$
4. **if** ($d_i ==$ **left)**
5. **for(** $j = 1$ **to** s **)** //seek from left to right in $children(T, g_r)$
6. **if** ($children(T, g_{root})[j]$ **==** t_i**)**
7. **Mark** $children(T, g_{root})[j]$ **as head**
8. **if** (d_i **== right)**
9. **for(** $j = s$ **to** 1 **)** //seek from right to left in $children(T, g_r)$
10. **if** ($children(T, g_r)[j] == t_i$ **)**
11. **Mark** $children(T, g_r)[j]$ **as head.**
12. **foreach** (T_k **in** $subtreeList(T)$)
13. $HF_R(T_k)$ //recursively call to subtrees

Fig. 3. A standard Head Finder Algorithm

Consider the tree in Figure 1. Suppose that the head-finder rule for tag VP is `((l TO)(l VBD)(l VBN)(l MD)(l VBZ)(l VB)(l VBG)(l VBP)(l VP)(l ADJP)(l NN)(l NNS)(l NP))`. When the head finder algorithm reaches the node VP, it looks from left to right a tag TO; since it cannot find such tag, it looks from left to right a tag VBD, it keeps changing what it is looking for until it looks from left to right for a tag VBZ. Once it has found it, it marks that subtree as head and it recursively inspects all subtrees.

3 A Quality Measure Based in Bilexical Grammars

This section introduces the quality measure used in our optimization procedure. Our procedure is based on the optimization of a quality measure q defined over a set of head-finder rules. In order to compute q for a given set of rules, we proceed as follows. We transform Sections 01-22 of PTB into dependency structures. Using the resulting dependency tree-bank we build a bilexical grammar, and finally, we compute a quality measure on this grammar. The measure over the bilexical grammar is a formula that takes into account PP and MS of the set of automata that define the bilexical grammar.

3.1 Bilexical Grammars

Bilexical grammars are a formalism in which lexical items, such as verbs and their arguments, can have idiosyncratic selective influences on each other. We define a *bilexical grammar* B as a 3-tuple $(R_o, \{r_c\}_{w \in C}, \{l_c\}_{c \in C},)$ where:

- C is a set of POS tags, C contains a distinguished symbol ROOT.
- For each tag $c \in C$, l_c and r_c there are two probabilistic automata with start symbols S_{l_c} and S_{r_c} respectively. Each automaton accepts some regular subset of C^*.

In contrast to the original definition where C is defined as a set of word, in ours, C is a set of POS tags. We use POS tags to reduce the complexity of building a bilexical grammar.

A *dependency tree* is a tree whose nodes (internal and external) are labeled with tags from C; the root is labeled with the symbol ROOT. The children ('dependents') of a

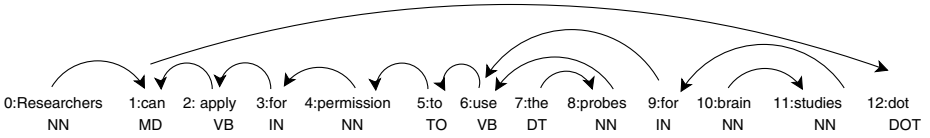


Fig. 4. Tree extracted from the PTB, file `wsj_0297.mrg` and transformed to a dependency tree

node are ordered in a sequence with respect to each other and the node itself, so that each node may have both *left children* that precede it and *right children* that follow it. A dependency tree T is *grammatical* if, for every tag token c that appears in the tree, l_c accepts the (possibly empty) sequence of c's left children (from right to left), and r_c accepts the sequence of c's right children (from left to right).

3.2 Induction of Bilexical Grammars

Bilexical grammars can be induced from a dependency tree-bank by inducing two automata for each tag in C. The induction of Bilexical Grammars is carried out in a supervised fashion. Our training material comes from transforming Sections 02–21 of the PTB into dependency trees. All words in the PTB are removed, leaving the POS tag as leaf in the dependency trees. We induce a bilexical grammar for each possible set of head-finder rules.

Once all trees in the tree-bank have been transformed to dependencies, we extract two bags T^c_{left} and T^c_{right} of *strings* for each POS c from the training material. An example illustrates the extraction procedure better: Figure 4 shows a dependency tree and Table 1 shows some of the bags of left and right dependents that are extracted. Note that in the example, all sets of strings displayed in the table are strings extracted only from the example tree. In the actual setting, T^c_{left} and T^c_{right} are built joining strings coming from all trees in the tree-bank.

Once T^c_{left} and T^c_{right} are extracted, two probabilistic automata A^c_{left} and A^c_{right} are built. For this purpose, we use the *minimum discrimination information* (MDI) [10] algorithm. The MDI algorithm receives as arguments a bag of strings and it outputs a probabilistic deterministic automata that accepts and generalizes over the input bag of strings. The algorithm has a unique parameter `alpha`, which we optimize during the grammar optimization phase as explained in Section 4. Since a bilexical grammar is defined through its automata, once all automata A^c_{left} and A^c_{right}, c in C are induced, the bilexical grammar associated to the tag set C is completely defined.

Table 1. Bags of left and right dependents extracted from dependency tree in Figure 4. Left dependents are to be read from right to left. All displayed sets are singletons.

Word #	i	T^i_{left}	T^i_{right}
0	NN	{NN}	{NN}
1	MD	{MD NN}	{MD VB DOTSYB}
2	VB	{VB}	{VB IN}
3	IN	{IN}	{IN NN}
4	NN	{NN}	{NN TO}

3.3 Quality Measure for Grammars

The measure q of a set of head-finder rules is defined as a measure of the grammar that is built from using the head-finder rule to transform the PTB into dependencies. The measure is then defined over the automata that defined the grammars. The measure over bilexical grammars contains two components. The first one, called *test sample perplexity* (PP), is the *per symbol log-likelihood* of strings belonging to a test sample according to the distribution defined by the automaton. The minimal perplexity $\mathrm{PP} = 1$ is reached when the next symbol is always predicted with probability 1, while $\mathrm{PP} = |\Sigma|$ corresponds to uniformly guessing from an alphabet Σ of size $|\Sigma|$. The second component is given by the number of *missed samples* (MS). A missed sample is a string in the test sample that the automaton fails to accept. One of such instance suffices to have PP undefined. Since an undefined value of PP only witnesses the presence of at least one MS we count the number of MS separately, and compute PP without considering MS. The test sample that is used to compute PP and MS comes from all trees in sections 00-01 of the PTB. These trees are transformed to dependency ones by using HF_{R_c}, where R_c is the candidate set of rules. Better values of MS and PP for a grammar mean that its automata capture better the regular language of dependents by producing most strings in the automata target languages with fewer levels of perplexity. The quality measure of grammar is then the mean of PP's and MS's for all automata in the grammar.

4 Building and Traversing the Search Space

This section introduces the search space and the strategy to traverse it in our optimization procedure. The search space consists of different sets of head rules. The standard set of rules contains 26 rules. The longest is the one associated with ADJP; it contains 18 entries. Finding a new set of rules means that we should find a new set of 26 vectors. For each candidate rule we have to transform the PTB, build the bilexical grammar, and compute PP and MS for all automata. It takes us 1.2 minutes to evaluate one candidate set of rules.

In principle, all possible head-rules can be candidate rules, but then the search space would be huge and it would be computationally unfeasible to traverse it. In order to avoid such search space, we run a series of experiments where we optimize one rule at a time. For example, one of our experiments is to optimize the rule associated with VB. Our search space contains all possible set of rules where all rules exept the one associated to VB are as in the standard set of head-finder rules.

To optimize one rule we traverse the search space with Genetic Algorithms. Genetic Algorithms need for their implementation (1) Definition of individuals: each individual codifies one candidate of the head-finder rule that is being optimize. (2) A fitness function defined over individuals: the quality measure is computed by constructing a set of head-finder rules by adding the candidate rule to the standard set of rules, building a bilexical grammatical and evaluating it as described in Section 3.3. Finally, (3) A strategy for evolution: we apply two different operations to individuals, namely crossover and mutation; crossover gets 0.95 probability of being applied, while mutation gets 0.05. We select individuals using the *roulette wheel strategy* [11]. In our experiments,

in each generation there is a population of 50 individuals; we let the population evolve for 100 generations.

The *mutation* function is easily defined by computing a random permutation of the rule tags vector and a random sample of its direction vector. The *crossover* operation is defined as follows. Let $[g_1, \ldots, g_i, \ldots, g_n]$ and $[h_1, \ldots, h_i, \ldots, h_n]$ be the tag vectors of two different individuals and let i be random number between 1 and n. The crossover produces two new individuals. The tag vector of one of the individuals is defined as follows:

$$sub([g_1, \ldots, g_n], [h_{i+1}, \ldots, h_n]) \cdot [h_{i+1}, \ldots, h_n]$$

where the operator $(\cdot)$ appends two arrays, and $sub(x, y)$ deletes the elements in x that are in y. The tag vector of the other individual is defined similarly by changing g by h and vice versa. The direction vectors of the new individuals are obtained by using the usual definition of crossover for boolean vector. In this way, crossover ensures that the resulting head rules do not have repeated tags.

5 Stability of Head Finders

As it is shown in Section 6, our optimization method only improves the performance of Bikel's parser. The reason for the lack of improvement of Stanford parser can be either because our optimization method is ill defined or because the parser is indifferent to the set of head-finders rules. However, using no head finder, that is, non dependency grammars, performance never reaches beyond 75%. So, heads and dependencies based on heads are an important element in parsing performance.

In this section we present experiments that try to shed light on this issue. We tested (1) randomly generated head finder rules, (2) head finders whose rules were reversed, (3) head finders that always choose right or left, and (4) inconsistent head finders.

Random Head Finders: We experiment generating several sets of head finder rules. A random set of head finder rules is created by replacing one rule in the standard set of head finder rules by one random rule. A random rule is created by randomly permuting the elements of both the tags vector and the direction vector. Experimental results for this head finder are shown in Table 3 (A).

In Figure 5, the first row shows the head rule defined in the standard set for category `S`. The second row shows a random permutation of this rule. The last row shows a reverse permutation of the original one. The reverse permutation of a rule is obtained by reversing the order of its *tag vector* and leaving its *direction vector* unchanged. In this example, the rule presented in the second row is calculated using the permutation $[7, 1, 4, 8, 2, 3, 6, 5]$ for the *tags vector* and its random *direction vector* was (l, r, r, l, l, l, l, l). We generate 119 rules and, consequently, 119 different head finder rule sets. Each of these sets differs in one rule from the standard set. In this way, we show the impact of each rule in the overall parsing performance.

We also experiment with a set of rules were *all* of its rules were randomly generated. We test 7 of such random sets for each parser. Experimental results for this head finder are shown in Table 4.

Original	`(S (1 TO)(1 IN)(1 VP)(1 S)(1 SBAR)(1 ADJP)(1 UCP)(1 NP))`
sampled rule	`(S (1 UCP)(r TO)(r S)(1 NP)(1 IN)(1 VP)(1 ADJP)(1 SBAR))`
reverse rule	`(S (1 NP)(1 UCP)(1 ADJP)(1 SBAR)(1 S)(1 VP)(1 IN)(1 TO))`

Fig. 5. The first row shows the original Collins's head rule for `S`. The second row shows a random permutation of the original rule. The last row is the reverse of the original rule.

Reverse Rules: There are 26 rules in the standard set of head finder rules. We generate 26 new sets by changing one rule at a time by its reverse. The reverse of a rule is constructed by reading it from left to right. The impact of reverse rules in parsing performance is shown in Table 3 (B).

Left Most and Right Most Head Finders: We define two special algorithms for finding heads. The always-leftmost and always-rightmost algorithm chooses for each internal node the leftmost and rightmost subtree respectively. These are special cases of head finder algorithms that cannot be expressed with a set of rules. In order to implement these algorithms, we modified both parser implementations. The results are shown in Table 2 (B).

Non-deterministic Head Finders: All previous experiments were based on deterministic head finders: every time they are used to transform a given phrase structure tree, they transform into the same dependency tree. We implemented a non-deterministic head finder algorithm, this algorithm flips a coin every time it has to decided where the head is. When this head finder is used to transform a phrase structure into a dependency tree it produces different dependency trees for every time it is called. We report results for 7 of these experiments for each parser, they can be seen in Table 4.

6 Experimental Results

In this section, we show the results of all our experiments. In all experiments we used Sections 02-21 of the PTB for training and Section 23 for testing. The optimization algorithm use Sections 00-01 for computing the quality measure defined on automata. Our experiments aim to analyze the variation in performance by changing one or more head rules in the standard set of head rules. The rules that are modified by our experiments correspond to tags: {`WHADJP, CONJP, WHNP, SINV, QP, RRC, S, ADVP, NAC, SBAR, VP, SQ, ADJP, WHPP, SBARQ, PP, WHADVP`}.

Table 2 (A) shows the performance of the set of rules produced by the optimization procedure. Each row displays labeled precision, labeled recall, significance level $pval$, and harmonic mean F_1. The baseline row reports the performance of both parsers using the standard set of rules. $pval$ was computed against the baseline. We consider a result as statistically significant if its significance level $pval$ is below 0.05. Performance value were computed using the `evalb` script, significance values were measured using Bikel's *Randomized Parsing Evaluation Comparator* script. The table shows that the performance in the Stanford parser using our optimized head finder set of rules is below the baseline, however, this decrease in performance is not statistically significant.

Table 2. (A) The result of the experiments corresponding to the optimized head finder. The upper part shows evaluation in Bikel's parser, while the bottom with Stanford parser. **(B)** First column shows the F_1 when all worst performing rules, reported in Table 4 (A), are put together. Second and third columns show average F_1 for the always right-most, and always left most head finders.

Num.	L. R.	L. P.	pval R.	pval P.	F_1
Bikel Baseline	88,53	88,63			88,583
optimal head	88,72	88,85	0,006	0,002	88,785
Stanford Baseline	85,26	86,23			85,742
optimal head	85,24	86,22	0,098	0,131	85,727

(A)

Parser	F_1 worst choice	F_1 Right Most	F_1 Left Most
Bikel	82,486	83,024	85,102
Stanford	84,092	84,206	85,566

(B)

The best set of head rules was obtained by combining all rules that our optimization method produced. Table 4 shows the results of the random head rule generation. The Table contains one row per each rule that was permuted. We consider 17 different rules, for each, we build $7 * 17$ new random rules. Each row shows the maximal, the minimal and the average F_1 measure we obtained.

Table 3 (B) shows F_1 measure for the 17 rules that were obtained by reversing one of the rules in the standard set at a time.

From Tables 3 (A) and (B) we can see that the rule defined for tag `VP` has the greatest impact on the performance of both parsers.

The first column of Table 2 (B) shows the results for the head finder that is built by using the 17 rules with the worst performance in experiments in Table 4. The second and the third columns show the results for the head finder algorithms that choose always the leftmost and always the rightmost respectively.

Table 3. (A) Parsing results obtained by replacing one rule in the standard set by a random rule. Each row shows the average, maximal and minimal impact in the F_1 measure for each parser. (B) Experiments result for each Head finder built with the reverse of head rule. One column for each parser.

	Bikel			Stanford		
rule tag	**avg.**	**max.**	**min.**	**avg.**	**max.**	**min.**
WHADJP	88,589	88,596	88,583	85,739	85,739	85,739
CONJP	88,586	88,601	88,582	85,739	85,739	85,739
WHNP	88,586	88,596	88,577	85,739	85,739	85,739
SINV	88,485	88,608	88,009	85,749	85,772	85,730
QP	88,538	88,604	88,474	85,747	85,759	85,732
RRC	88,588	88,595	88,583	85,739	85,739	85,739
S	88,092	88,569	87,458	85,689	85,726	85,652
ADVP	88,586	88,615	88,564	85,740	85,743	85,739
NAC	88,594	88,607	88,581	85,743	85,743	85,743
SBAR	88,195	88,653	88,013	85,734	85,739	85,733
VP	**87,330**	**88,471**	**85,870**	**85,247**	**85,612**	**84,918**
SQ	88,571	88,592	88,562	85,739	85,739	85,739
ADJP	88,616	88,698	88,566	85,727	85,739	85,718
WHPP	88,583	88,583	88,583	85,739	85,739	85,739
SBARQ	88,583	88,583	88,583	85,739	85,739	85,739
PP	88,617	88,668	88,583	85,740	85,740	85,739
WHADVP	88,607	88,706	88,583	85,739	85,739	85,739

(A)

Gram.tag	F_1^B	F_1^S
WHADJP	88,596	85,739
SBAR	87,990	85,733
CONJP	88,600	85,739
VP	85,820	84,249
WHNP	88,596	85,739
SQ	88,562	85,739
SINV	88,465	85,758
ADJP	88,626	85,726
QP	88,490	85,748
WHPP	88.583	85,739
RRC	88,595	85,739
SBARQ	88.583	85,739
S	88,178	85,670
PP	88.583	85,739
ADVP	88,613	85,739
WHADVP	88.593	85,739
NAC	88,603	85,743

(B)

Table 4. Experiments result of random choice of rules, for each experiment we show the impact in the F_1 measure for the average, maximal and minimal

	Rand. No Det.			Random Det		
Parser	**avg.**	**max.**	**min.**	**avg.**	**max.**	**min.**
Bikel	86,976	87,166	86,754	86,001	87,974	83,857
Stanford	84,810	84,997	84,691	84,805	85,625	84,360

Table 4 shows results for the non-deterministic and deterministic head finders. The set of rules are obtained by changing rules for the 17 tags considered in our work. We run 7 tests for deterministic and 7 tests for non-deterministic head finders. In both cases, we calculate the average, the maximum and the minimum, obtained for the measure F_1. The results show that the non-deterministic head finder is more stable because the variation between the minimum and the maximum results is lower. A priori, this is a surprising result, because the dependency trees used to induce the grammar during the training phase have percolated inconsistent heads. We think that the non-deterministic head finders are more stable because, in average, they make more "correct" choices. In contrast, if deterministic head finders contain an erroneous rule, all the resulting dependency trees are wrong. This fact is also supported by the results reported in Table 2 (B). It shows that using the head finder built out of the worst performing rules is the one with the worst performance in both parsers. The performance drops nearly 6% and 1.9% for Bikel and Stanford respectively. The right-most head finder decline is the next considering the performance downfall.

7 Related Work

Similar work has been published in [12], and an improved version can be found in the Bikel's thesis [4]. In this work, the authors tried to induce head rules by means of defining a generative model that starts with an initial set of rules and uses an EM-like algorithm to produce a new set of rules that maximize the likelihood. They used different sets of rules as seeds for the EM, but the approach only shows improvement when the standard set of rules is used. In contrast to our approach, none of their improvements were statistically significant. They also show that when the seed is a set of random rules, the overall performance decreases.

In a different approach [13] the authors present different unsupervised algorithms for head assignments used in their Lexicalized Tree Substitution Grammars. They study different types of algorithms based on entropy minimization, familiarity maximization, and several variants of these algorithms. Their results shows that using the head finder they induced, they obtain an improvement of 4% over a PCFG parser using an standards head assignments. In our work, we don't use lexicalized grammars. Our approach is based on improvements to a given rule set, as opposed to theirs where they use unsupervised methods to find assigments for heads.

8 Conclusions

In our approach, we aim at generating dependency trees that improve the performance of the statistical parser. To do so, we vary the head rules that are used while transforming

constituent trees into dependency ones. Besides finding some new rules which hardly improve parsers performance, we found that variations in head finding algorithms do not have a decisive impact on the syntactic parsers performance to the extent that an aleatory translation of the phrase structure trees into dependency ones can be used without damaging considerably the parsing performance. However, removing head finding altogether produces a 10% decrease in performance, considerably higher than the $1,9\%$ and 6% decreases in performance produced by the worst possible head finders. Therefore, head finders are crucial for the performance of dependency parsers but their variations are not.

References

1. Magerman, D.M.: Natural language parsing as statistical pattern recognition. Ph.D. thesis, Stanford University (1994)
2. Charniak, E.: A maximum-entropy-inspired parser. In: NAACL 2000 (2000)
3. Klein, D., Manning, C.: Accurate unlexicalized parsing. In: Proc. 41st ACL (2003)
4. Bikel, D.: On the Parameter Space of Generative Lexicalized Statistical Parsing Models. PhD thesis, University of Pennsylvania (2004)
5. Collins, M.: Three generative, lexicalized models for statistical parsing. In: ACL 1997 (1997)
6. Eisner, J.: Bilexical grammars and a cubictime probabilistic parser. In: Proceedings of IWPT04 (1994)
7. Klein, D., Manning, C.: Distributional phrase structure induction. In: CoNLL 2001 (2001)
8. Nivre, J.A.A.: Maltparser: A language-independent system for data-driven dependency parsing. In: Natural Language Engineering, pp. 95–135 (2007)
9. Marcus, M., Santorini, B.: Building a large annotated corpus of English: The Penn treebank. Computational Linguistics 19, 313–330 (1993)
10. Thollard, F., Dupont, P., de la Higuera, C.: Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In: Proc. ICML, Stanford (2000)
11. Gen, M., Cheng, R.: Genetic Algorithms and Engineering Design. John Wiley, Chichester (1997)
12. Chiang, D., Recovering, D.B.: latent information in treebanks. In: COLING 2002, Taipei, Taiwan (2002)
13. Sangati, F., Zuidema, W.: Unsupervised methods for head assignments. In: EACL, pp. 701–709 (2009)

Estimating the Birth and Death Years of Authors of Undated Documents Using Undated Citations

Yaakov HaCohen-Kerner[1] and Dror Mughaz[2,1]

[1] Dept. of Computer Science, Jerusalem College of Technology, 91160 Jerusalem, Israel
[2] Dept. of Computer Science, Bar-Ilan University, 52900 Ramat-Gan, Israel
kerner@jct.ac.il, myghaz@cs.biu.ac.il

Abstract. Precious historical treasures might be hidden between the lines of a text. There are many implicit details which can be extracted from a text, particularly if one has access to an entire corpus of texts pertaining to the given subject. One of these details is the identification of the era in which the author of the given document(s) lived. For rabbinic documents written in Hebrew and Aramaic, which are almost without exception undated and do not contain any bibliographic section, this problem is extremely important. The aim of this novel research is to find in which years an author was born and died, based on his documents and the documents of other authors (whose birth and death years are known) who refer to the author under discussion or are mentioned by him. Such estimates can help determine the time frame in which certain documents were written and in some cases identify an anonymous author. In the framework of this research, we formulate various kinds of "iron-clad", heuristic and greedy constraints defining the birth and death years of an author based on citations referring to him or mentioned by him. Experiments applied on a corpus containing texts composed by rabbinic authors show reasonable results.

Keywords: Citation analysis, Hebrew, Hebrew-Aramaic documents, knowledge discovery, time analysis, undated citations, undated documents.

1 Introduction

Citations are a defining feature of many kinds of documents, e.g., academic, legal and religious. Authors cite previous works which are related in some way to their own work or to their discussion. Citations included in documents are important information resources of interest to researchers. Therefore, automatic extraction and analysis of citations from documents are of great importance.

Recent developments (e.g., computerized corpora and search engines) enable accurate extraction of citations. As a result, citation analysis has an increased importance.

A citation is a brief reference in the body of the text to a source of published information. A reference includes bibliographic details about a source that is mentioned in a citation. The reference is found at end of a document in a reference list. Citations are presented in agreed typographical formats. Different disciplines have different conventions: citation in footnotes, citations with numbers (e.g., [1]) or mixed symbols such as [Cohen98] or [Cohen 1998] (Harvard-style citations).

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 138–149, 2010.

Garfield [2] was the first to propose automatic production of citation indexes, extraction and analysis of citations from corpora of academic papers. Powley and Dale [5] develop techniques to extract from a given academic paper a list of citations and, for each citation, the corresponding reference in the reference list. They find each instance of a citation in the body of the paper; parse it into a set of author names and years; and find the segment of text from the references which contains the corresponding reference.

Teufel et al. [8] use extracted citations and their context for automatic classification of citations to their citation function (the author's reason for citing a given paper). Some research has been done concerning the improvement of retrieval performance using terms. Ritchie et al. [6] show that document indexing based on combinations of terms used by citing documents and terms from the document itself give better retrieval performance than standard indexing of the document terms alone. In [7], Ritchie et al. investigate how to select text from around the citations in order to extract good index terms in order to improve retrieval effectiveness.

Citations are a defining feature not just of academic papers but also and even more of rabbinic responsa (answers written in response to Jewish legal questions authored by rabbinic scholars). Citations included in rabbinic literature are more complex to define and to extract than citations in academic papers written in English because:

(1) In contrast to academic papers, there is no reference list that appears at the end of a responsa;

(2) There is an interaction with the complex morphology of Hebrew and Aramaic. For example, citations can be presented with different types of prefixes (e.g., "and …", "when …", "and when …", "in …", "and in …", "and when in …") included in the citation-word(s);

(3) Natural language processing in Hebrew and Aramaic has been relatively little studied;

(4) Many citations in Hebrew-Aramaic documents are ambiguous. For instance: (a) a book titled מגן-אבות *magen-avot* was composed by four different Jewish authors; and (b) The abbreviation מ"ב (m"b) relates to two different Jewish authors and has also other meanings, which are not authors' names; and

(5) At least 30 different syntactic styles are used to present citations. This number is higher than the number of citation patterns used in academic papers written in English (e.g., see [5]).

Each specific document written by a specific author can be referred to, in at least 30 general possible citation syntactic styles. Furthermore, each citation pattern can be expanded to many other specific citations by replacing the name of the author and/or his book/responsa by each one of their other names (e.g., different spellings, full names, short names, first names, surnames, and nicknames with/without title) and abbreviations.

The citation recognition in this research is done by comparing each word to a list of 298 known authors and many of their books/responsa. This list contains 19,506 specific citations that relate to names, nick names and abbreviations of these authors and their writings. Basic known citations were collected and all other citations were produced from them, based on an automatic extension process using regular expressions.

Hebrew-Aramaic documents in general and Hebrew-Aramaic responsa in principle present various interesting text mining problems. Firstly, Hebrew is richer in its morphology forms than English. According to linguistic estimates, Hebrew has 70,000,000 valid (inflected) forms while English has only 1,000,000 [1]. In Hebrew, there are up to seven thousand declensions for one stem, while in English there are only a few declensions. Secondly, these kinds of documents include a high rate of abbreviations (about 20%), while more than one third of them (about 8%) are ambiguous [4].

A previous research that works on corpora, which contain responsa referring to Jewish law written in Hebrew-Aramaic dealt with text classification [3]. In this research, HaCohen-Kerner et al. investigate whether the use of stylistic feature sets and/or name-based feature sets is appropriate for classification of documents to the ethnic group of their authors and/or periods of time when the documents were written and/or places where the documents were written. In addition, HaCohen-Kerner et al. [4] have experience with the processing of such texts from the viewpoint of disambiguation of ambiguous abbreviations. The current research is a continuation of this long-term research interest.

In this research, we present a novel model that estimates the birth and death years of a given author using undated citations of other authors (whose birth and death years are known) who refer to him or mentioned by him. The documents are undated (non-time-stamped) and mentions of years or historical events in the documents are very rare. The estimations are based on various constraints of different degree of certainty: "iron-clad", heuristic and greedy constraints. The constraints are based on general citations without cue words and citations with cue words, such as father, son, rabbi, teacher, student, friend, and "late" ("of blessed memory").

This paper is organized as follows: Section 2 presents various constraints of different degree of certainty: "iron-clad", heuristic and greedy constraints that are used to estimate the birth and death years of responsa authors. Section 3 describes the model. Section 4 introduces the tested dataset, the results of the experiments and their analysis. Section 5 summarizes, concludes and proposes future directions.

2 Citation-Based Constraints

This section presents the citation-based constraints formulated for the estimation of the birth and death years of an author X based on his documents and on other authors' (Y_i) documents who mention X or one of his documents. We assume that the death years (for those who died) and birth years of all authors are known, excluding those of the investigated author. Below are given some notions and constants that are used: X – The author under consideration, Y_i – Other authors, B – Birth year, D – Death year, MIN – Minimal age (currently 30 years) of a rabbinic author when he starts to write his response, MAX – Maximal life period (currently 100 years) of a rabbinic author, and MIN_FATHER – Minimal age (currently 20 years) of a rabbinic author when his firstborn son is born.

The estimations of MIN, MAX and MIN_FATHER constants are only heuristic, although they are realistic on the basis of typical responsa authors' lifestyle.

Various types of citations exist: general citations without cue words and citations with cue words, such as: father, son, rabbi, teacher, student, friend, and "late" ("of blessed memory"). Another classification of the discussed citations is to those referring to living authors and those referring to dead authors. In contrast to academic papers, responsa include much more citations to dead authors than to living authors.

We will introduce citation-based constraints of different degrees of certainty: "iron-clad" (I), heuristic (H) and greedy (G). "Iron-clad" constraints are absolutely true, without any exception. Heuristic constraints are almost always true. Exceptions can occur when the heuristic estimates for MIN, MAX and MIN_FATHER are incorrect. Greedy constraints are rather reasonable constraints for responsa authors. However, sometimes wrong estimates can be drawn while using these constraints. Each constraint will be numbered and its degree of certainty will be presented in brackets.

2.1 "Iron-clad" and Heuristic Constraints

First of all, we present two general heuristic constraints based on authors that cite X, which are based on regular citations (i.e., without mentioning special cue words, e.g., friend, son, father and rabbi).

General constraint based on authors that were cited by X

$$D(X) >= MAX(B(Yi))+MIN \quad (1\ (H))$$

X must be alive when he cited Y_i, so we can use the earliest possible age of publishing of the latest born author Y as a lower estimate for X's death year.

General constraint based on authors that cite X

$$B(X) <= MIN(D(Yi))-MIN \quad (2\ (H))$$

All Y_i must have been alive when they cited X, and X must have been old enough to publish. Therefore, we can use the earliest death year amongst such authors Y_i as an upper estimate of X's earliest possible publication age (and thus his birth year).

Posthumous citation constraints

Posthumous constraints estimate the birth and death years of an author X based on citations of authors who refer to X as "late" ("of blessed memory") or on citations of X who mentions other authors as "late". Figure 1 describes possible situations where various kinds of authors Y_i (i=1, 2, 3) refer to X as "late". The lines depict authors' life spans where the left edges represent the birth years and the right edges represent death years. In this case (as all Y_i refer to X as "late"), we know that all Y_i died after X (and some of the Y_i might be still alive), but we do not know when they were born in relation to X's birth. Y_1 was born before X's birth; Y_2 was born after X's birth but before X's death; and Y_3 was born after X's death.

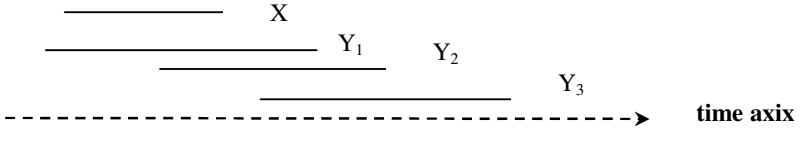


Fig. 1. Citations mentioning X as "late"

$$D(X) <= MIN(D(Yi)) \quad (3\ (I))$$

However, we know that X must have been dead when Y_i cited him as "late", so we can use the earliest born such Y's death year as an upper estimate for X's death year. Like all authors, dead authors of course have to comply to constraint (2) as well.

Let us now look at the cases where the author X, we are studying refers to other authors Yi as "late". Figure 2 describes possible situations where X refers to various kinds of authors Y_i (i = 1, 2, 3) as "late". All Y_i died before X's death (or maybe X is still alive). Y_1 died before X's birth; Y_2 was born before X's birth and died when X was still alive; and Y_3 was born after X's birth and died when X was still alive.

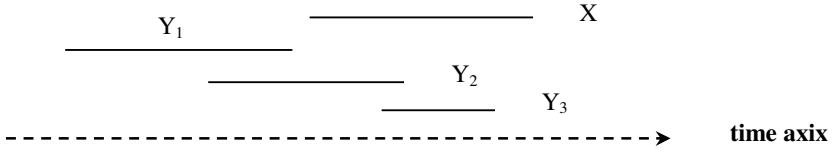


Fig. 2. Citations by X who mentions others as "late"

$$D(X) >= MAX(D(Yi)) \quad (4\ (I))$$

X must be alive after the death of all Y_i who were cited as "late" by him. Therefore, we can use the death year of the latest-born such Y as a lower estimate for X's death year.

$$B(X) >= MAX(D(Yi)) - MAX \quad (5\ (H))$$

X was probably born after the death year of the latest-dying person, who X wrote about. Therefore, we can use the death year of the latest-born such Y minus his maximal life-period as a lower estimate for X's born year.

Contemporary citation constraints

Contemporary citation constraints calculate the upper and lower bounds of the birth year of an author X based only on citations of known authors who refer to X as their friend/student/rabbi. This means there must have been at least some period in time when both were alive and intellectually active. Figure 3 describes possible situations where various kinds of authors Y_i refer to X as their friend/student/rabbi. Y_1 was born before X's birth and died before X's death; Y_2 was born before X's birth and died after X's death; Y_3 was born after X's birth and died before X's death; and Y_4 was born after X's birth and died after X's death. Like all authors, contemporary authors of course have to comply to constraints 1 and 2 as well.

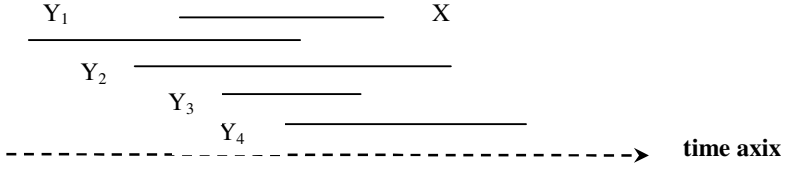


Fig. 3. Citations by authors who refer to X as their Friend/Student/Rabbi

$$B(X)>= MIN(B(Yi))-(MAX-MIN) \quad (6\ (H))$$

All Y_i must have been alive when X was alive, and all of them must have been old enough to publish. Therefore, X could not be born MAX-MIN years before the earliest birth year amongst all authors Y_i.

$$D(X)<=MAX(D(Yi))+(MAX-MIN) \quad (7\ (H))$$

Again, all Y_i must have been alive when X was alive, and all of them must have been old enough to publish. Thus, X could not be alive MAX-MIN years after the latest death year amongst all authors Y_i.

Intellectual son/father-based constraints

Son-based constraints calculate the upper and lower bounds of the birth and death years of an author X based only on citations of only one known author who refers to X as his son. According to rabbinic conventions, X can be either a "truly son" (i.e., a biological son), or an "intellectual son" (i.e., a student).

Figure 4 describes five possible situations. Y_i (i = 1, 2, 3) refer to X as their "truly son". In all these cases, Y_i were born before X's birth. Y_1 died before X's birth (maximum 9 months before X's birth); Y_2 died before X's death; and Y_3 died after X's death. Y_1 is not a possible father in the discussed context, since in this case, Y_1 cannot refer to his son who was born only after Y1's death. However, in Jewish rabbinic documents, it is possible that an author Y_i (e.g. Y_4 or Y_5) will call his student X, a son (meaning an intellectual son), although X is not his "truly son". In such a case, Y_i (the "father") can be born even after X's birth.

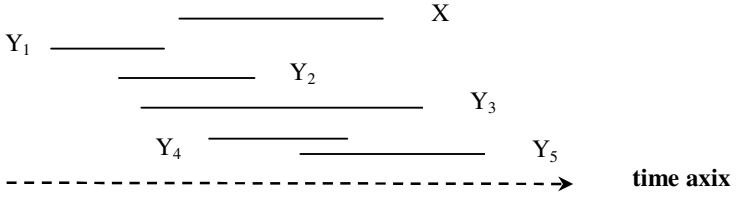


Fig. 4. Citations by authors who refer to X as their son

When taking into account situations such as an intellectual son (X towards Y_4 or Y_5), all son-based constraints are expressed by the friend/student/rabbi-based constraints (6-7). If a biological bond, i.e., a "truly son" can be absolutely identified, than a unique constraint can be formulated.

Father-based constraints calculate the upper and lower bounds of the birth and death years of an author X based only on citations of known authors who refer to X as their father. Also here, according to rabbinic conventions, X can be either a "truly father" (i.e., a biological father), or an "intellectual father" (i.e., a rabbi or a teacher). Therefore, all father-based constraints are expressed by the friend/student/rabbi-based constraints (6-7).

2.2 Greedy Constraints

We also formulate and apply greedy constraints. These bounds are sensible in many cases, but which can nevertheless sometimes lead to wrong estimates. It is important to mention that the greedy constraints are applied in combination with the iron-clad and heuristic constraints. This is because, in many cases some of the greedy constraints are not applied because lack of explicit citations (citations with cue words). In such cases, we use the estimations that are products of the iron-clad and heuristic constraints.

Greedy constraint based on authors who are mentioned by X

$$B(X) >= MAX(B(Yi)) \quad (8\ (G))$$

Most of the citations in our research domain, relate to dead authors. Thus, most of the citations mentioned by X relate to dead authors. That is, most of Y_i were born before X's birth and died before X's death. Therefore, a greedy assumption will be that X was born no earlier than the birth of latest author mentioned by X.

Greedy constraint based on authors who refer to X

$$D(X) <= MIN(D(Yi)) \quad (9\ (G))$$

As mentioned above, most of the citations mentioned by Y_i relate to X as dead. Therefore, most of Y_i die after X's death. Therefore, a greedy assumption will be that X died no later than the death of the earliest author who refers to X.

Refinement of constraints (8-9) are presented by constraints (10- 13). Constraints (10-11) are due to X citing Y_i and Constraints 12-13 are due to Y_i citing X.

Greedy constraint for defining the birth year based only on authors who were cited by X

$$B(X) >= MAX(D(Yi)) \quad (10\ (G))$$

When taking into account only citations that are cited by X, most of the citations, relate to dead authors. That is, most of Y_i died before X's birth. Therefore, a greedy assumption will be that X was born no earlier than the death of the latest author mentioned by X.

Greedy constraint for defining the birth year based only on authors who are mentioned by X as a friend

$$B(X) <= MIN(B(Yi)) \quad (11\ (G))$$

When taking into account only citations that are mentioned by X, which relate to contemporary authors, a greedy constraint can be that X was born no later than the birth of the earliest author mentioned by X as a friend.

Greedy constraint for defining the death year of X based only on authors who cited X as "late"

$$D(X) <= MIN(B(Yi)) \quad (12\ (G))$$

When taking into account only citations that are mentioned by Y_i who relate to X as "late", a greedy assumption can be that X died no later than the birth of the earliest author who cited X as "late".

Greedy constraint for defining the death year of X based only on authors who cited X as a friend

$$D(X) >= MAX(D(Yi)) \quad (13\ (G))$$

When taking into account only citations that are mentioned by Yi who cited X as a friend, all Y_i must have been alive when X was alive, and all of them must have been old enough to publish. Therefore, a greedy assumption will be that X died no earlier than the death of the latest author who cited X as a friend.

We do not present greedy constraints regarding son and father because they can be intellectual son and father and not truly relatives.

3 The Model

The main steps of the model are presented below. Most of these steps were processed automatically, except for steps 2 and 3 that were processed semi-automatically.

1. **Cleaning the texts.** Since the responsa may have undergone some editing, we must make sure to ignore possible effects of differences in the texts resulting from variant editing practices. Therefore, we eliminate all orthographic variations.
2. **Normalizing the citations in the texts.** For each author, we normalize all kinds of citations that refer to him (e.g., various variants and spellings of his name, books, documents and their nicknames and abbreviations). For each author, we collect all citation syntactic styles referred to him and then replace them to a unique string.
3. **Building indexes**, e.g., authors, citations to "late"/friend/student/rabbis/son/father and calculating the frequencies of each item.
4. **Citation identification** into various categories of citations, including self-citations.
5. **Performing various combinations of "iron-clad" and heuristic constraints** on the one hand, **and greedy constraints** on the other hand, **to estimate** the birth and death years for each tested author.
6. **Calculating averages and std-deviations** for the best "iron-clad" and heuristic version and the best greedy version.

4 Experimental Results

The examined dataset includes 3,488 responsa[1] authored by 12 Jewish rabbinic scholars, two of whom are still alive. All these authors lived in the last 130 years and were very productive regarding the number of documents and citations that were written by them. On average, there are about 291 documents for each scholar. These responsa were written in the last 100 years. The total number of words is about 6,887,351 words (average per documents is 1,975 words). This corpus includes citations to 298

[1] Contained in the Global Jewish Database (The Responsa Project at Bar-Ilan University). Http://www.biu.ac.il/ICJI/Responsa

authors including the 12 investigated authors. The dataset before the normalization step (step # 2 in section 4) includes 106,923 citations (i.e., mentions of other works), which are about 8,910 citations in average for each author and about 31 citations for each document. 19,506 of these citations are different.

Since this dataset represents a special corpus containing responsa authored by 12 authors who lived in the last 130 years, the incoming posthumous citations count is always 0. This special situation enables us to correct death ages which are higher than the current year. That is, if the upper bound of D(X) is greater than the current year then we change it to the current year. If the investigated authors died a few hundreds years ago, then the upper bounds would probably been much worse.

The situation with these authors also means that we did not apply average posthumous constraints (greedy rules # 8, 10 for the birth year and greedy rules # 9, 12 for the death year). In a different corpus situation (where all authors are roughly from the same period), these greedy rules help, but not here, where many ancient authors are cited (i.e., some of the lower bounds can be hundreds years ago and if we use them than the estimation for B(X) will be too low and therefore very bad).

Several characteristics of this dataset are presented below:

On average, each author cites 8,910 citations while only about 10 of them are posthumous citations and about 6 of them are contemporary citations. About 99.8% of the citations are implicit, i.e., they are not accompanied with cue words that identify whether the citations are posthumous or contemporary.

The average number of citations to each author is 88 including self citations and 33 excluding self citations. That is, most of the citations (62.5%) are self citations.

Among the explicit citations (those with cue-words) the average number of posthumous citations (10.25) is about twice greater than the average number of contemporary citations (5.67). That is, about two-thirds of the explicit citations are posthumous.

On average, for each author there are much more outgoing citations (8,910) than incoming citations (88) in general and more outgoing contemporary citations (6) than incoming contemporary citations (4).

Table 1 compares the ground truth about the birth and death years on the one hand to the best iron-clad and heuristic version and on the other hand to the best greedy version.

Since this is a novel problem, it is difficult to evaluate the results in the sense that although we can compare how close the system guess is to the actual birth/death years, what we cannot do is assess how-close-is-close, i.e. there is no real notion of what a 'good' result is.

Currently, we use the notion difference, which is defined as the estimated value minus the ground truth value. Some of the estimates for birth and death years are not integer values. This finding is due to the use of average functions in certain versions (e.g., two last sub-rows in tables 2 and 3).

Table 1 shows that the best experimental results have been achieved by the best greedy version, which was better than the best iron-clad and heuristic version as follows: (1) Its average birth-year and death-year differences (13.04 and 15.54, respectively) are better than those of the best iron-clad and heuristic version (22 and 22.67, respectively), (2) The absolute differences of 12 out of 24 estimates were less or equal to 6.5 years, versus only 5 such estimates of the best iron-clad and heuristic version and (3) The standard deviation of the birth-year's greedy estimate is less than its comparable

iron-clad and heuristic standard deviation. This indicates that the results of the best greedy version are steadier.

Indeed, the best greedy version was better than the best iron-clad and heuristic version only in 14 out of 24 estimates (of birth and death years). Therefore, these results are still not enough significant.

Table 2 presents the experimental results using the various iron-clad and heuristic constraints only (section 2.1). The minimal average birth-year and death-year differences (22 and 22.67, respectively) have been achieved by the version of the average "late"-based constraints (constraints 3-6). This result was obtained using the average

Table 1. Experimental results using various groups of constraints

	Author X	Ground truth		Best iron-clad & heuristic version		Differences for best iron-clad & heuristic version		Best greedy version		Differences for best greedy version	
#	Name of X (in Hebrew)	Birth year	Death year	Birth year	Death year	Birth year	Death year	Birth year	Death year	Birth year	Death year
1	אליעזר וולדינברג	1917	2006	1879	1971.5	38	34.5	1899.5	1953	17.5	53
2	בצלאל שטרן	1911	1989	1885	1959.5	26	29.5	1910	1989	**1**	**0**
3	עובדיה יוסף	1920	Alive	1888.5	1981	31.5	29	1894	1953	26	57
4	בן-ציון עוזיאל	1880	1953	1862.5	1952	17.5	**1**	1884	1959.5	**-4**	**-6.5**
5	יצחק הרצוג	1889	1959	1888.5	1981	**0.5**	-22	1874.5	1958	14.5	**1**
6	יצחק וויס	1902	1989	1887	1958.5	15	30.5	1880.5	1995	21.5	**-6**
7	יעקב עדס	1898	1963	1857.5	1950	40.5	13	1885	1980.5	13	-17.5
8	משה פיינשטיין	1895	1986	1913.5	1988	-18.5	**-2**	1889	1959	**6**	27
9	עובדיה הדאיה	1890	1969	1833.5	1923	56.5	46.5	1889	1971.5	**1**	**-2.5**
10	רחמים חוויתה	1901	1959	1915.5	1980.5	-14.5	-21.5	1874.5	1950	26.5	9
11	שמואלווזנר	1914	Alive	1916	1981	**-2**	29	1920	2009	**-6**	**1**
12	שלמה זלמן אויערבך	1910	1995	1906.5	1981	**3.5**	14	1890.5	1989	19.5	**6**
					Ave.	**22**	**22.67**		**Ave.**	**13.04**	**15.54**
					Std. dev.	17.15	13.28		Std. dev.	9.32	20.00

Table 2. Experimental results using different groups of constraints

Group of cons.	Upper and lower bounds	Average of absolute differences (in years)	
		Birth year	Death year
Cons. 1-2	B(X)< , D(X)>	35.83	38.67
Posthumous citation cons. (cons. 2-3)	B(X)< , D(X)<	43.42	26.33
	B(X)< , D(X)>	43.42	55.83
	B(X)> , D(X)<	43.42	26.33
	B(X)> , D(X)>	75.75	55.83
(cons. 2,5) & (cons. 3,4)	B(X)=ave(B(X)<, B(X)>)	22.00	
	D(X)=ave(D(X)<, D(X)>)		22.67
Contemporary cons. (cons. 1-2, 4-5)	B(X)< , D(X)<	37.58	33.67
	B(X)< , D(X)>	37.58	38.25
	B(X)> , D(X)<	87.58	33.67
	B(X)> , D(X)>	87.58	38.25
	B(X)=ave(B(X)<, B(X)>)	45.08	29.79
	D(X)=ave(D(X)<, D(X)>)		

of the upper and the lower bounds of the birth year as estimate for the birth year and the average of the upper and the lower bounds of the death year as estimate for the death year. This version is better than the version that contains the two most simple constraints (1-2), which do not take into consideration any cue-words. This finding indicates that the posthumous and contemporary constraints do contribute to the estimates.

The result achieved by the best iron-clad version was successful also because an important correction that was done by us concerning the iron-clad constraints dealing with the estimation of D(X). That is, if the upper bound of D(X) is greater than the current year then we change it to the current year. If the investigated authors died a few hundreds years ago, then the upper bounds would probably been much worse. In general, the results achieved by the contemporary (friend) constraints were worse than those achieved by the "late" constraints. That might be due to the fact that there more posthumous citations than contemporary citations.

Table 3. Experimental results using different versions of the greedy constraints

Group of cons.	Average of absolute differences (in years)	
	Birth year	Death year
Cons. 8-9	13.42	17.30
Posthumous cons. (10, 12)	43.42	26.30
Contemporary cons. (1-2, 4-5)	37.58	33.67
Ave. friend cons. B(X) = ave(10,11), D(X) = ave(12,13)	13.04	15.54

Table 3 presents the results achieved by the different versions of the greedy constraints (section 2.2). The minimal averages of absolute differences (in years) for the birth and death years (13.04 and 15.54, respectively) have been achieved by the greedy version of the average "friend"-based constraints (constraints 10-13).

5 Summary, Conclusions and Future Work

To the best of our knowledge, we are the first to investigate the estimation of the birth and death years of the authors using undated citations referring to them or written by them. This investigation was performed on a special case of documents (i.e., responsa), where special writing rules are applied. The estimation was based on the author's documents and documents of other authors (whose birth and death years are known) who refer to the discussed author or are mentioned by him. To do so, we formulate various kinds of iron-clad, heuristic and greedy constraints. The best estimates have been achieved using the version of the average contemporary greedy constraints.

Regarding the estimation of the birth and death years of an author X, it is important to point that citations mentioned by X or referring to X are more suitable to estimate the "birth" and "death" writing years of X rather than his real birth and death years.

This model might be applied with suitable changes to similar research problems that might be relevant for some historical legal or religious document collections. Usually, such documents include citations to previous documents of the same kind.

We plan to improve the estimation of the birth and death years of authors by: (1) Combining and testing new combinations of iron-clad, heuristic and greedy constraints, (2) Improving existing constraints and/or formulating new constraints (e.g., statistical-based constraints), (3) Defining and applying heuristic constraints that take into account various details included in the responsa, e.g., dates (in case that they appear), events, names of people, concepts, special words and collocations that can be dated, (4) Conducting additional experiments using many more responsa written by more authors is supposed to improve the estimates, (5) Checking why the iron-clad, heuristic and greedy constraints tend to produce more positive differences, and (6) Testing how much of an improvement we got from a correction of the upper bound of D(x) and how much we will at some point use it for a corpus with long-dead authors.

Definition and application of additional kinds of constraints is planned: (1) Constraints that are based on historical events mentioned in the documents; and (2) Three-generation constraints, i.e., constraints that relate to biological or preceding relations, e.g., grand son and grand student. Another interesting future research is the disambiguation of ambiguous citations.

Acknowledgements. The authors thank Simone Teufel for reviewing drafts of this article and offering many helpful comments, and three anonymous reviewers for their reviews.

References

1. Choueka, Y., Conley, E.S., Dagan, I.: A Comprehensive Bilingual Word Alignment System: Application to Disparate Languages - Hebrew, English. In: Veronis, J. (ed.) Parallel Text Processing, pp. 69–96. Kluwer Academic Publishers, Dordrecht (2000)
2. Garfield, E.: Can Citation Indexing be Automated? In: Stevens, M. (ed.) Statistical Association Methods for Mechanical Documentation, Symposium Proceedings, vol. 269, pp. 189–142. National Bureau of Standards Miscellaneous Publication (1965)
3. HaCohen-Kerner, Y., Beck, H., Yehudai, E., Rosenstein, M., Mughaz, D.: Cuisine: Classification using Stylistic Feature Sets and/or Name-Based Feature Sets. To appear in Journal of the American Society for Information Science and Technology, JASIST (2010) (Published Online: Apr 22 2010), (DOI: 10.1002/asi.21350)
4. HaCohen-Kerner, Y., Kass, A., Peretz, A.: HAADS: A Hebrew Aramaic Abbreviation Disambiguation System. To appear in Journal of the American Society for Information Science and Technology, JASIST (2010) (Published Online: May 27, 2010), (DOI: 10.1002/asi.21367)
5. Powley, B., Dale, R.: Evidence-Based Information Extraction for High Accuracy Citation and Author Name Identification. In: RIAO 2007 (2007)
6. Ritchie, A., Teufel, S., Robertson, S.: Using Terms from Citations for IR: Some First Results. In: The European Conference for Information Retrieval (ECIR), pp. 211–221 (2007)
7. Ritchie, A., Robertson, S., Teufel, S.: Comparing Citation Contexts for Information Retrieval. In: The 17th ACM Conference on Information and Knowledge Management (CIKM), pp. 213–222 (2008)
8. Teufel, S., Siddharthan, A., Tidhar, D.: Automatic Classification of Citation Function. In: The 2006 Conference on Empirical Methods in Natural Language Processing, ACL, pp. 103–110 (2006)

Using Temporal Cues for Segmenting Texts into Events

Ludovic Jean-Louis, Romaric Besançon, and Olivier Ferret

CEA LIST, Vision and Content Engineering Laboratory,
Fontenay-aux-Roses, F-92265 France
{ludovic.jean-louis,romaric.besancon,olivier.ferret}@cea.fr

Abstract. One of the early application of Information Extraction, motivated by the needs for intelligence tools, is the detection of events in news articles. But this detection may be difficult when news articles mention several occurrences of events of the same kind, which is often done for comparison purposes. We propose in this article new approaches to segment the text of news articles in units relative to only one event, in order to help the identification of relevant information associated with the main event of the news. We present two approaches that use statistical machine learning models (HMM and CRF) exploiting temporal information extracted from the texts as a basis for this segmentation. The evaluation of these approaches in the domain of seismic events show that with a robust and generic approach, we can achieve results at least as good as results obtained with a specialized heuristic approach.

Keywords: Information extraction, text segmentation, temporal cues.

1 Introduction

The detection of events has always been a major issue of Information Extraction. It was already addressed in the *Message Understanding Conferences* (MUC) [6] and is still a subject of interest in more recent evaluations such as ACE (Automatic Content Extraction) [4]. This detection is the trigger of the extraction process which aims at filling the slots of a template defining the typical information associated with a certain type of events. In the domain of terrorist attacks for instance, such process identifies the type of an attack (bombing, etc.) and extracts slot information such as its date, its location or its target. The content of these slots are typically named entities whereas events appear as a direct relation between named entities (for example, the <Date>September 1989</Date> <Hurricane>**Hugo Hurricane**</Hurricane>), as a verb or a verbal noun (<Hurricane>Hurricane Hugo</Hurricane> **struck** <Location>South Carolina</Location> in <Date>1989</Date>), or can extend beyond the scope of a sentence. Most of the work about slot filling in Information Extraction focuses on the first two cases, mainly because the identification of a relation between the mention of an event and a named entity often use lexico-syntactic patterns or syntactic relations. However, as it was already underlined in [11] and later analyzed more

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 150–161, 2010.

precisely in [22], a significant number of such relations can be identified only at the discourse level. This fact was taken into account in some existing works mainly through coreference resolution [23] or by acquiring and using domain-knowledge for guiding the slot filling process [21,8].

In this article, we tackle this issue through the means of discourse segmentation. More specifically, we propose to segment texts according to the events they refer to in order to narrow the span of text to explore for linking a named entity to an event mention. As time is an important feature for discriminating events, as illustrated by [18], we chose to perform this segmentation by relying on temporal cues.

The extraction of temporal information from texts has been widely studied in different fields of Natural Language Processing, since this kind of information is useful for many applications. In the field of Information Extraction, temporal information is for instance use to find the ordering of events [5,16] or to identify their overlapping [18]. In our work, the dates associated with event mentions are useful for segmenting texts into events. However, each mention of an event doesn't necessarily appear with a date in the same sentence as it is illustrated by the following example:

(1) *<Hurricane> Hurricane Hugo</Hurricane> in <Date>1989</Date> was a <Level>class 4</Level> storm. ...*

(2) *<Hurricane>Hurricane Hugo</Hurricane> caused <Damages>$7 billion damage</Damages> in the Caribbean Sea and South Carolina.*

In sentence 1, the date *1989* is linked to the event mention through syntactic dependencies whereas, without named entity coreference resolution, linking a date to the event mention in sentence 2 is not possible. The problem of associating a date to an event is addressed for instance in [5], which proposes a set of heuristics for assigning time-stamps to the events of a text by relying on three temporal cues: in each sentence, the presence of a date and the tenses of verbs; more globally, the date of the document. In [7], each sentence is not assigned a date but a date associated with an event is propagated to an undated event according to the relations between them (*e.g.*, a cause-consequence relation).

In relation to this problem, [15] proposes a segmentation model that views texts as sequences of situations. These situations are defined through three types of entities: *temporal entities, locations and persons*. Moreover, [15] distinguishes between texts with a simple structure and texts with a complex structure. The first ones are centered on one event only that is considered from one viewpoint. All entities in this case tend to contribute to the definition of this event. The second type of texts refer to several events among which a main event can be distinguished together with subordinate events.

In this article, we focus more particularly on the segmentation into events of texts with a complex structure. We present the general principles of our work in Section 2 while Section 3 is dedicated to the way we apply these principles with machine learning techniques. Finally, we present in Section 4 the results of the evaluation of the method on French news articles in the seismic domain.

2 Principles and Objectives

Event extraction as presented in this article takes place in a wider context of technology watch in which users are mainly interested in the most recent events. In this context, our goal is to synthesize from news articles the information about such recent events into a kind of dashboard[1]. However, news articles often refer to several comparable events, generally for pointing out the similarities and differences between a recent event and older ones. In our case, we are not interested in these older events and we consider them as a source of noise for extracting information about the main event of a news article. We adopted a two-step strategy to tackle this problem:

- segmenting texts according to the events they refer to. These segments are frequently discontinuous as the structure of news articles is often dominated by moves between their main event and past similar events;
- linking to the event of a segment the named entities that are part of its features.

This strategy is illustrated by Figure 1 with a text about a seismic event.

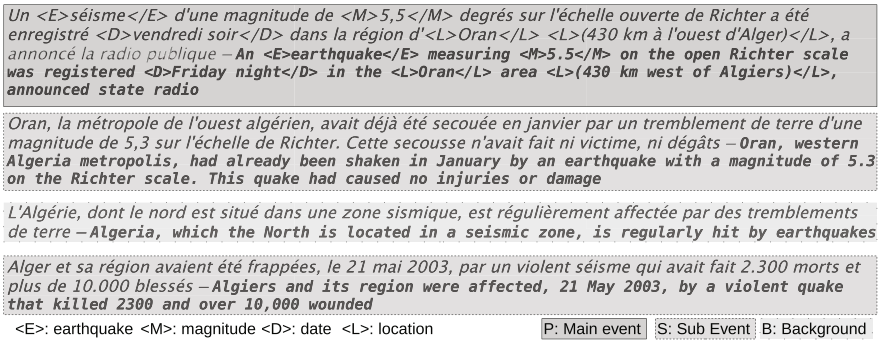


Fig. 1. Segmentation of a text into events

The work of this article mainly focuses on the segmentation of texts into events. Following this perspective, we adopted a representation of texts turned towards events: a text is viewed as a sequence of sentences in which each sentence is characterized by the presence or the absence of an event[2]. In addition, we also focus in this work on the identification of the named entities associated with the main event of a news article. We have therefore decided not to differentiate one secondary event from another. Thus, we propose to classify sentences according to the following three categories:

[1] This approach is a little bit different from most works in the information extraction field in which information is searched for all the events of a certain type.

[2] The hypothesis "one sentence = one event" is a simplification but it is globally not too simplistic in our application domains.

- **Main event:** all sentences referring to the main event of the text;
- **Secondary event:** all sentences containing data that are related to an event different from the *Main Event*;
- **Background:** all sentences that belong neither to the *Main Event* or a *Secondary Event*.

To perform this classification, we assume that the most interesting criteria rely not only on the nature of sentences but also on their linking at a discursive level, with the idea that categories of events don't follow one another in a arbitrary way. At the linguistic level, the transition from one category of event to another one can be marked by several types of cues: the presence of a temporal marker (typically, a date) introducing a new temporal reference or the use of a grammatical tense indicating a return to the past for instance. In the example of Figure 1, the sequence of tenses *present perfect/past perfect/present/past perfect* corresponds to the sequence of types of events *main/secondary/background/secondary*. This sequence of events is also marked by the presence of a date in the second and the last sentences, which refer to a secondary event, and by an expression in the third sentence, *regularly*, that expresses temporal recurrence. More globally, our aim is to capture the dependencies between the shifts of temporal frames and the shifts of events through machine learning methods to segment news articles into events.

3 Approaches for Segmenting Text into Events

In this work, we treat the problem of segmenting texts into events as a classification problem: each sentence of the text must be associated with an event type. As we also consider that the sequence of sentences contain valuable information for this classification, and we want classify all sentences, a graphical model of sequence annotation is particularly suited. We describe in this section two standard sequence classification models for this task: Hidden Markov Model (HMM) and Conditional Random Fields (CRF).

3.1 Text Preprocessing

The segmentation itself uses a sentence representation composed of linguistic cues extracted from the text. We therefore perform a linguistic analysis of the text, consisting in tokenization, sentence boundary detection, Part-Of-Speech tagging, verb tense analysis and named entity recognition. This linguistic processing pipeline is implemented using the linguistic analyzer LIMA [1].

3.2 HMM Model

Hidden Markov Model [19] is a sequence classifier widely used in NLP (for example, named entity recognition, POS-tagging etc) and has also been applied on text segmentation (in particular for topic segmentation [24]). HMM are stochastic models used to find an underlying sequence of hidden states from the sequence

of observable data. In our case, we want to find the sequence of events associated with a given text, considered as a sentence sequence.

We assume that the segmentation is a Markov process, which means that the state associated with the current sentence only depends on previous state: we propose to use the temporal information (verb tenses) as observation and the event categories as hidden states. The transition matrices are obtained from a corpus of manually annotated texts (supervised learning). An illustration of the HMM model used in our approach is presented in Figure 2.

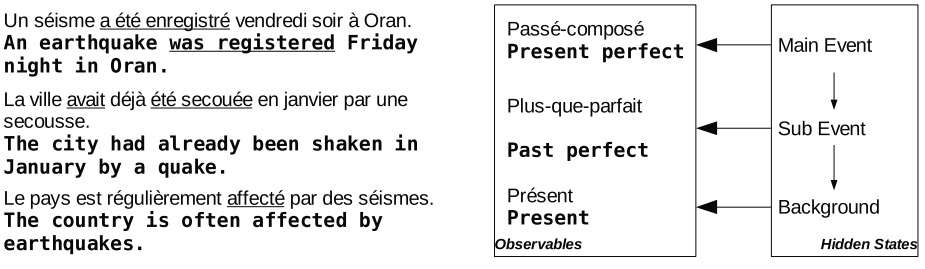


Fig. 2. Illustration of text segmentation using HMM

A constraint of the HMM is that for a given observation, the sequence of states does not originally take into account dependencies between the current state and previous observations and does not allow to easily integrate several criteria (without multiplying the number of observation and the amount of training data). To avoid this constraint, we also tested a CRF model, presented in the following section.

3.3 CRF Model

Since they were introduced in 2001, Conditional Random Fields [13] have been widely and successfully used in several NLP fields. In text segmentation and classification, [9] have obtained good results applying CRF models to classify sentences contained in scientific abstracts into four categories : *objective, methods, results, conclusions.*

The main difference between HMM and CRF is that HMM aim at maximizing the joint probability $P(x, y)$ between a sequence of observations (x) and a sequence of state (y) whereas CRF compute conditional probability $P(y|x)$ in order to associate a sequence of states with the observations. The advantage of the conditional approach is that it allows to represent the sequence of observation as a vector whose components are related to features attached to the observation. These features allow to integrate more linguistic expert knowledge in the models. A more formal definition of CRF is:

$$P(y|x) = \frac{1}{Z_\lambda(x)} exp(\lambda.F(y, x))$$

$$F(y,x) = \sum_i f(y,x,i) \qquad Z_\lambda(x) = \sum_y exp(\lambda.F(y,x))$$

where $F(y,x)$ is a vector whose components are the values of the features for the input sequence, and λ is a vector of weights associated with the features and $Z_\lambda(x)$ is a normalizing factor which depends on all possible sequences of states.

To apply CRF to text segmentation, we use the following features:

- **verb tenses:** we make the hypothesis that changes of verb tenses, in particular concerning past tenses, are correlated to changes of event types in news articles. We take into account this dimension in the CRF by associating a binary feature with each possible verb tense (feature is 1 if at least one verb of the sentence is at the considered tense);
- **presence of dates:** if a sentence contains a date (*i.e.* if the author has considered necessary to specify a date) it is probable that it refers to an event different from the previous sentence (except for the first occurrence of a date). We exploited this idea with the integration of a binary feature related to the presence/absence of a named entity of type DATE in the sentence (in the current model, the value of the date is not used);
- **temporal expressions:** this feature takes into account the presence of temporal expressions in a sentence, such as *over the past two weeks, in recent years*, often related to generalities. We use a dictionary of such expressions in French, manually built from the annotated corpus from [14].

4 Evaluation

We present in this section the results obtained by applying statistical learning algorithms for segmenting texts into events. The evaluation process is composed of two stages: first, an intrinsic evaluation of the segmentation approach ("are the identified segments correct?") and second, a final evaluation on the intended application, *i.e.* the impact of text segmentation on the extraction of the entities associated with the main event. We compare the statistical-based approaches to a domain specific heuristic (*HeurSeg*) and a paragraph-based heuristic (*ParaSeg*). *HeurSeg* is used in an existing application dedicated to information extraction for seismic events. It is mainly based on the presence and the value of dates: different date values correspond to different text segments (the main event segment being the one containing the most recent date) and between two distinct dates, the boundaries of the segment are set according to the presence of sentence and paragraph breaks and the presence of domain-specific entities between the two dates. *ParaSeg* determines event categories at a paragraph level: a sentence is associated with the main event category if it belongs to the first two paragraphs; otherwise, the sub event category is chosen.

We notice that HMM and CRF use different features for their classification decisions: HMM decision is only based on the sequence of verb tenses whereas CRF decision uses a richer set of features (described in section 3.3). Therefore, we also use a Maximum Entropy model (MaxEnt [20]) as comparative model for the CRF, with the same set of features, in order to confirm the interest of the information given by the sequence for the segmentation.

From an implementation point of view, we used a set of Python scripts together with several reference toolkit: NLTK[3], CRF++[4] and MaxEnt[5] respectively for HMM, CRF and MaxEnt models.

4.1 Corpus

In order to evaluate our text segmentation approach, we used a corpus of 501 French news articles concerning earthquake events. The articles were collected between February 2008 and early September 2008, partly from the French *Agence France Presse* newswire (1/3 of the corpus), and partly from Google News (2/3 of the corpus). These articles deal with 142 distinct major earthquake events. The corpus contains both articles with a simple structure (only one event) and articles with a complex structure (several events): 252 articles (50%) report at least one secondary event. The corpus has been manually annotated for named entities by domain analysts, only for entities associated with the article main event[6]. Table 1 reports the categories of named entities associated with an earthquake event together with their distribution in the corpus. The table shows that the distribution of named entities is not homogeneous: many location names are found (947) whereas only few geographical coordinates (Geo) are present in the articles (30). As a consequence, the overall performance of the latter category will be strongly influenced by a matched/missed entity while the former entity category will not. In order to evaluate our text segmentation approach we annotated a subset of the corpus (composed of 140 articles) with event segmentation information. Most of the selected articles in the subset contain a main and a secondary event (some short articles might not contain secondary event information). Table 2 shows the distribution of event categories in the annotated subset. The most represented event category is *Main Event (70%)*, which is consistent with the factual aspect of news articles. The *Secondary Event* includes without distinction all earthquake events that are comparable to the *Main Event*: notice that among the selected news the real number of distinct secondary events evoked in the article can go up to 4. Finally, the annotated subset has on average 1.7 distinct secondary events mentioned per article.

4.2 Intrinsic Evaluation of the Segmentation of Texts into Events

This section presents the results in terms of precision and recall percentage (denoted P and R) for different text segmentation approaches. These results where obtained through 5-fold cross validation on the annotated subset (140 news articles), using 4/5 for training and the remaining 1/5 for testing. We

[3] http://www.nltk.org/
[4] http://crfpp.sourceforge.net/
[5] http://webdocs.cs.ualberta.ca/ lindek/downloads.htm
[6] Annotators could annotate several entities of the same type if they thought the entities were equally satisfactory as pieces of information related to the article main event (for instance for location entities, annotators could annotate both a city and a country name).

Table 1. Distribution of named entities in the annotated corpus: 3306 named entities for 501 articles

Entity type	Number	Nature
Event type	499	type of event (earthquake, tsunami...)
Location	947	place where the event occurred
Date	470	date when the event occurred
Time	345	time when the event occurred
Magnitude	484	magnitude
Damages	531	damages caused by the event
Geo	30	geographical coordinates of the event

Table 2. Distribution of the event categories in the annotated corpus

1659 annotated segments in 140 news articles		
Event type	Number	Percentage of annotated segments
Main Event	1168	70%
Secondary Event	287	17%
Background	213	13%

report in Table 3 the segmentation results for statistical and heuristic-based approaches. We first notice that both *ParaSeg* and *HeurSeg* are outperformed for the precision on Main Event by assigning all sentences to the most frequent category (see Table 2). Results for *ParaSeg* and *HeurSeg* are comparable in terms of precision but *ParaSeg* achieves a poor score on recall, which is explained by the fact that most articles contain short paragraphs and few sentences are associated with main event in this case.

Concerning the statistical segmentation models, the HMM results prove that the unique verb tense criterion is not sufficient to discriminate all event categories: while the main event category is correctly identified (83.0% recall and 93.6% precision), secondary event and background categories are poorly identified (and they can be useful for a general purpose Information Extraction task). Results also demonstrate that performances for secondary event and background categories are improved with CRF and MaxEnt models (better results with CRF), which confirms the importance of considering the successive sentence categories while segmenting the text into events.

4.3 Evaluation of Text Segmentation for Information Extraction

The goal of the text segmentation presented here is to delimit text segments that refer to a single event category in order to link the relevant entities to the main event (linking is made within a text segment). In order to evaluate the impact of the segmentation on this linking task, we used a simple heuristic for linking the entities to the main event, based on the hypothesis that pieces of information contained in the articles are organized according to their importance in the newscast: the most relevant pieces of information (which are generally associated

Table 3. Results for segmentation into events

	ParaSeg		HeurSeg		HMM		MaxEnt		CRF	
Event type	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*
Main Event	6.1	63.9	82.8	64.7	83.0	93.6	94.8	78.7	98.7	87.4
Secondary Event	86.9	12.4	23.5	43.4	37.8	9.6	33.6	54.7	52.6	95.8
Background	0.0	0.0	16.9	21.7	49.1	40.0	22.0	84.2	69.3	93.0

Table 4. Results for the entity to main event linking

	NoSeg		ParaSeg		HeurSeg		HMM		CRF		Gold Std	
Entity type	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*
Damages	83.5	77.9	62.1	61.0	76.3	74.4	69.8	65.1	80.2	75.3	76.7	73.5
Date	38.4	35.9	34.3	32.6	69.3	65.0	48.9	45.6	64.4	60.1	89.5	86.9
Event type	82.1	81.6	78.7	78.2	79.3	78.8	59.1	58.8	76.7	76.2	85.6	85.6
Geo	86.7	96.3	44.8	50.0	66.7	74.1	86.7	96.3	83.3	92.6	100.0	100.0
Location	41.0	40.9	39.2	39.1	56.0	55.9	61.2	61.1	57.4	57.3	86.4	86.4
Magnitude	93.5	93.0	72.7	73.2	86.2	85.9	66.7	66.2	86.7	86.1	93.4	93.4
Time	61.0	51.2	25.3	22.5	56.4	49.2	78.8	71.5	63.4	55.5	92.2	91.3
All	66.6	63.5	53.1	51.8	71.0	68.6	63.4	61.1	71.7	68.8	87.5	86.3

with the main event) generally appear before the subordinate pieces of information (associated with either a secondary event or the background). Following this idea, we use the following linking heuristic: *for each entity category, select as most relevant entity the first entity found in the segment.*

Table 4 summarizes the results of the entity linking task while using heuristic-based and learning-based approaches for the segmentation into events. We also report as baseline the results of the entity linking without event segmentation (*NoSeg*) and with the reference segmentation on the 140 annotated articles (*Gold Std*). *NoSeg* achieves fairly good results for the entity linking process (even higher than the HMM: +2.7% F1-Measure). These results are significantly improved by *HeurSeg* (+6.2% F1-Measure compared to *NoSeg*), which demonstrates the interest of segmentation for the entity linking task. Even if there are variations depending on the type of entities, the CRF model gives results equivalent to those obtained with the heuristic approach (and even slightly better), which is quite satisfactory since the CRF model is a generic method while the heuristic is explicitly dependent on the domain. Finally, the best results obtained by *Gold Std* also show that there is still room for improvement using a smarter entity linking strategy.

5 Related Work

The work we have presented in this article focuses on text segmentation, with two main characteristics: this segmentation is performed according to the event

type of sentences and relies on temporal cues. The use of temporal cues for discourse segmentation was mainly explored by linguistic and psycho-linguistic works through the study of the role of clause-initial temporal adverbials as segmentation markers. From the psycholinguistic viewpoint, [2] showed that clause-initial temporal adverbials are correlated with topic shifts whereas from a more linguistic viewpoint, [10] uncovered a more complex situation where the role of clause-initial temporal adverbials is text-type dependent. Our use of temporal cues for text segmentation is both more crude and more extensive: we mainly rely on the sequence of grammatical tenses and we use other temporal cues, such as the presence of dates or temporal adverbials, as features for identifying more accurately secondary event and background sentences.

Segmenting texts according to their events was also addressed by some few works. In [12], this segmentation was mainly based the identification in texts of the components of a type of events, defined by *a priori* domain knowledge. However, two typical discourse structures of the news articles they considered were also taken into account for this segmentation: one is made of a sequence of different events while the other one is structured as in our case round a main event with several mentions with references to minor events. [3] also heavily based its segmentation of texts on the identification the components of *a priori* template but tested several discourse-inspired heuristics for assigning a clause to an event as for instance favoring the event of the most recently assigned clause. Finally, the closest work to ours from this viewpoint is [17], which tagged sentences with four event labels: new event, continuing event, back-reference to an earlier event, no reference to an event. These labels are close to our three event types, except that we don't distinguish the introduction of a new event from its continuation. The model used in [17] is a probabilistic Finite State Automaton relying on the MDI algorithm for its learning part. While this work aims at modelling the discourse structure of texts from the viewpoint of events, it differs from ours in that it directly models sequences of event labels and don't exploit temporal information. [17] showed that its segmentation had a positive impact on its final task, *i.e.* grouping sentences in news articles that refer to the same event, but didn't report results for only segmentation.

6 Conclusion and Future Work

The aim of our work is to segment texts into events in order to make easier the linking of relevant entities to the main event of a news article. In our approach, we addressed this segmentation problem as a sequence classification task where the goal is to determine the type of event associated with each sentence. We proposed and evaluated three models: a HMM model, which uses as single decision criterion the succession of tenses in a text, a MaxEnt model which integrates additional temporal cues (temporal expressions, dates) into its decision, and a CRF model which integrates the same features as MaxEnt in addition to the dependencies between the successive event types. While conducting comparative experiments of the models on a corpus of news articles concerning earthquake

events, we have shown that the CRF model provides the best results for the segmentation of texts into events. In addition, we tested the impact of segmenting the texts into events on the identification of the entities related to the main event and we showed that CRF, which is a learning-based model, achieves equivalent results (and even slightly better) than those obtained with a heuristic-based approach, by using a much more generic approach.

Regarding the generalization of the approach, we have obtained encouraging results for initial tests by applying the models on an English corpus of news articles, using the models learned from French. Concerning the application domain, we used a corpus of news articles related to the earthquake field in which information is quite structured. Nevertheless, we believe our approach can provide reasonable results in other areas, which we are planning to experiment in the near future. Finally, a detailed error analysis on the test corpus showed that the process of linking the entities to the main event is a major source of error as we are currently using a simplistic heuristic. We will therefore focus our future research on this issue by using both entity density criteria and linguistic criteria (*e.g.* explicit syntactic dependencies between entities).

References

1. Besançon, R., de Chalendar, G., Ferret, O., Gara, F., Laib, M., Semmar, N.: LIMA: A Multilingual Framework for Linguistic Analysis and Linguistic Resources Development and Evaluation. In: 7^{th} Conference on Language Resources and Evaluation (LREC 2010), Malta (2010)
2. Bestgen, Y., Vonk, W.: Temporal Adverbials as Segmentation Markers in Discourse Comprehension. Journal of Memory and Language 42(1), 74–87 (2000)
3. Crowe, J.: Constraint-based Event Recognition for Information Extraction. In: 33^{rd} Annual Meeting of the Association for Computational Linguistics (ACL 1995), Cambridge, Massachusetts, USA, pp. 296–298 (1995)
4. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R.: The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. In: 4^{th} Conference on Language Resources and Evaluation (LREC 2004), pp. 837–840 (2004)
5. Filatova, E., Hovy, E.: Assigning Time-Stamps to Event-Clauses. In: ACL Workshop on Temporal and Spatial Information Processing, France, pp. 1–8 (2001)
6. Grishman, R., Sundheim, B.: Message Understanding Conference-6: A Brief History. In: 16^{th} International Conference on Computational linguistics (COLING 1996), Copenhagen, Denmark, pp. 466–471 (1996)
7. Gupta, P., Ji, H.: Predicting Unknown Time Arguments Based on Cross-Event Propagation. In: ACL-IJCNLP 2009 (short papers), Singapore, pp. 369–372 (2009)
8. Harabagiu, S.: Incremental Topic Representations. In: 20^{th} International Conference on Computational Linguistics (COLING 2004), Switzerland, pp. 583–589 (2004)
9. Hirohata, K., Okazaki, N., Ananiadou, S., Ishizuka, M.: Identifying Sections in Scientific Abstracts using Conditional Random Fields. In: Third International Joint Conference on Natural Language Processing (IJCNLP 2008), Hyderabad, India, pp. 381–388 (2008)

10. Ho-Dac, L.M., Péry-Woodley, M.P.: Temporal adverbials and discourse segmentation revisited. In: 7^{th} International Workshop on Multidisciplinary Approaches to Discourse 2008 (MAD 2008) - Linearisation and Segmentation in Discourse, Lysebu, Oslo, Norway, pp. 65–77 (2008)
11. Iwańska, L., Appelt, D., Ayuso, D., Dahlgren, K., Stalls, B.G., Grishman, R., Krupka, G., Montgomery, C., Riloff, E.: Computational aspects of discourse in the context of MUC-3. In: DARPA (ed.). MUC-3, pp. 256–282 (1991)
12. Kitani, T., Eriguchi, Y., Hara, M.: Pattern Matching and Discourse Processing in Information Extraction from Japanese Text. Journal of Artificial Intelligence Research 2, 89–110 (1994)
13. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Eighteenth International Conference on Machine Learning (ICML 2001), USA, pp. 282–289 (2001)
14. Laporte, E., Nakamura, T., Voyatzi, S.: A French Corpus Annotated for Multiword Expressions with Adverbial Function. In: 6^{th} Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Maroc, pp. 48–51 (2008)
15. Lucas, N.: The enunciative structure of news dispatches, a contrastive rhetorical approach. In: Language, Culture, Rhetoric, ASLA (Association Suédoise de Linguistique Appliquée), pp. 159–164 (2005)
16. Mani, I., Verhagen, M., Wellner, B., Lee, C.M., Pustejovsky, J.: Machine Learning of Temporal Relations. In: 21^{st} International Conference on Computational Linguistics and 44^{th} Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006), Sydney, Australia, pp. 753–760 (2006)
17. Naughton, M.: Exploiting Structure for Event Discovery Using the MDI Algorithm. In: 45^{th} Annual Meeting of the Association for Computational Linguistics (ACL 2007), Prague, pp. 31–36 (2007)
18. Pustejovsky, J., Knippen, R., Littman, J., Sauri, R.: Temporal and Event Information in Natural Language Text. Computers and the Humanities 39(2-3), 123–164 (2005)
19. Rabiner, L.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Readings in Speech Recognition, 267–290 (1989)
20. Ratnaparkhi, A.: Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. thesis, Philadelphia, PA, USA (1998)
21. Soderland, S., Lehnert, W.: Wrap-Up: a Trainable Discourse Module for Information Extraction. Journal of Artificial Intelligence Research 2, 131–158 (1994)
22. Stevenson, M.: Fact distribution in Information Extraction. Language Resources and Evaluation 40(2), 183–201 (2006)
23. Surdeanu, M., Harabagiu, S.M.: Infrastructure for Open-Domain Information Extraction. In: Second International Conference on Human Language Technology Research (HLT 2002), San Diego, California, pp. 325–330 (2002)
24. Yamron, J.P., Carp, I., Gillick, L., Lowe, S., van Mulbregt, P.V.: A Hidden Markov Model Approach to Text Segmentation and Event Tracking. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 333–336 (1998)

Enriching the Adjective Domain in the Japanese WordNet

Kyoko Kanzaki[1], Francis Bond[2], Takayuki Kuribayashi[1], and Hitoshi Isahara[1]

[1] National Institute of Information and Communications Technology,
3-5 Hikaridai, Seikacho, Sorakugun, Kyoto 619-0289, Japan
{kanzaki,kuribayashi,isahara}@nict.go.jp
[2] Nanyang Technology University
14, Nanyang Drive, Singapore 637332
bond@ieee.org

Abstract. We released Japanese WordNet Version 1.0 in March 2010, and are continuing to enrich the Japanese WordNet in several directions. The current version of the Japanese WordNet is a kind of translation of Princeton WordNet 3.0 and we used WordNets of multiple languages in order to disambiguate Japanese translations. Although the structure is based on Princeton WordNet 3.0, some information is still insufficient in our Japanese WordNet. For example, the adjective domain lacks semantic relations such as antonyms, attributes and so on. As part of our ongoing work to enrich the Japanese WordNet, we are investigating attribute nouns for which adjectives express values.

Keywords: WordNet, Adjective, Japanese.

1 Japanese WordNet

We are building a Japanese version of WordNet, and released version 1.0 in March 2010. The WordNet project at Princeton has been a resounding success, creating a resource that is widely used in research and has been emulated in many languages: [7]. In order for a lexical resource to be widely adopted it must be both accessible and usable. The Princeton WordNet is accessible because it is released under a nonrestrictive license; and it is usable because it has not just precise information but also reasonable coverage, especially of common words.

There have been several initiatives to create a Japanese WordNet, but none of them has yet produced something that is both accessible and usable. The large amount of previous work shows the great interest and value of producing a Japanese WordNet. We therefore decided to construct one as follows: [1]. First, automatically translate the Princeton WordNet into Japanese. All relations (e.g., hypernyms and meronyms) come from Princeton WordNet 3.0. Second, the most frequent 20,000 synsets are manually checked. Third, the synsets are linked to a corpus. Fourth, the data are released under an open license.

The Japanese WordNet is based on the structure of the English WordNet: Japanese near synonyms are added to the existing English synsets. For example, one of the English synsets consisting of “seal” has the explanation “any of numerous marine

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 162–166, 2010.

mammals that come on shore to breed; chiefly of cold regions", and has the following Japanese words associated with it: "アザラシ *azarashi*" and "海豹 *azarashi*".

A statistical summary of the contents of WordNet 3.0 and Japanese WordNet 1.0 is shown in Table 1.

Table 1. Statistics for WordNet 3.0 and Japanese WordNet 1.0 (from websites [9] and [10])

	WordNet 3.0	Japanese WordNet 1.0
Unique strings	155,287	92,241
Synsets	117,659	56,741
Senses word-sense pair in English synset-word pair in Japanese	206,941	157,398

2 Creating the Japanese WordNet

Our approach to building the Japanese WordNet is the standard expansion approach: "translate WordNet synsets to another language and take over the structure" [8]. We did this both to keep a compatible structure with WordNet, and because we had access to a variety of resources that would make the task easier.

Our main innovation is that we are using WordNets in multiple languages to disambiguate the Japanese translations, thus providing more reliable estimates. The English-Japanese lexicon has two translations for "bat", i.e. "蝙蝠 *koumori* (mammal)" and "バット *batto* (club)". However, because there is no way of distinguishing between them, we get a mixture of meanings with "蝙蝠 *koumori* (bat#n#1)" and "バット*batto* (bat#n#5)". "Chiropteran (bat#n#1)" is not in any of the English-Japanese lexicons we used, and bat#n#5 has no synonyms. Therefore, using only the English WordNet as source and English-Japanese lexicons, there is no way to disambiguate them.

However, both synsets are also in the French WordNet: bat#n#1 is "chauve-souris" and bat#n#5 is "batte (gourdin)". These are not ambiguous in the same way: *chauve-souris* goes only to "蝙蝠 *koumori*" and *batte* only to "バット*batto*". Thus, if we can match through two languages, the mapping is much more likely to be the correct sense.

The actual algorithm we used was as follows:

For each synset in WordNet 3.0:
- Find equivalents in WordNets of French, Spanish and German.
- Look up all Japanese translations via English, French, Spanish and German WordNets.
- Rank Japanese equivalents.

  score s = |links| + 10 for links in two languages

The result is a WordNet with multiple Japanese candidates for most synsets, with a confidence score s equal to the number of bilingual links plus a ten-point bonus for

being linked in multiple languages. Thus we built and released Japanese WordNet 1.0, but many improvements remain to be made.

3 Adjective Domain in WordNet 3.0 and Japanese WordNet

As for the adjective domain, basic adjectives were translated in our Japanese WordNet. Table 2 shows a statistical comparison of the adjective domain between WordNet 3.0 and Japanese WordNet. However, the adjective domain in the Japanese WordNet is still insufficient compared to WordNet 3.0.

Table 2. A statistical comparison of the adjective domain

	Unique strings	Synsets	Word-sense pairs
WordNet 3.0	21,479	18,156	30,002
Japanese WordNet	4,494	8,915	17,679

According to WordNet 3.0, adjectives are arranged in clusters containing head synsets and satellite synsets. Each cluster is organized around antonymous pairs (and occasionally antonymous triplets). The antonymous pairs (or triplets) are indicated in the head synsets of a cluster. Most head synsets have one or more satellite synsets, each of which represents a concept that is similar in meaning to the concept represented by the head synset. In the Japanese WordNet, head synsets and satellite synsets are not clearly distinguished.

As shown in the following examples, adjectives in WordNet 3.0 have several semantic pointers such as <antonym>, <attribute>, <similar to>, <derivationally related form>, <also see> and <pertainym>, by which adjectives are related with other synsets (i.e. not only other adjectives but also other parts of speech like nouns and verbs). In the Japanese WordNet, they have not been created yet. These useful relations between synsets or words across a part of speech should be introduced in the Japanese WordNet.

Examples 1: ("< >" refers to a pointer, and "[ ]" refers to a synset)

Pointer <antonym>
01661289-a: [good, right, ripe] –<antonym>– 01661914-a: [inopportune]
Pointer <attribute>
01498769-a: [measurable, mensurable] –<attribute>– 05090441-n: [magnitude]

4 Enriching the Adjective Domain in the Japanese WordNet – Detecting Attribute Concepts and Their Values –

In WordNet 3.0, through the pointer <attribute>, an adjective is linked to a noun for which an adjective expresses its value. Such nouns, called attributive nouns, are linked to their hypernyms via a pointer <inherited hypernym>. As an example, hypernyms for "abundant" are shown below. The hypernyms "quantity", "amount",

"magnitude", "property", "attribute", "abstraction" and "entity" can be identified via the pointer <attribute> and <inherited hypernym>.

Example 2: An example of inherited hypernym for "abundant"

abundant –<attribute>– quantity
quantity –<inherited hypernym> (from direct hypernym to upper hypernym)–
S: (n) amount (the relative magnitude of something with reference to a criterion) "an adequate amount of food for four people"
S: (n) magnitude (the property of relative size or extent (whether large or small)) "they tried to predict the magnitude of the explosion"; "about the magnitude of a small pea"
S: (n) property (a basic or essential attribute shared by all members of a class) "a study of the physical properties of atomic particles"
S: (n) attribute (an abstraction belonging to or characteristic of an entity)
S: (n) abstraction, abstract entity (a general concept formed by extracting common features from specific examples)
S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Also, another pointer <see also> is similar to the pointer <attribute> in terms of linking an adjective to a noun, but it is considered to be a more indirect related word.

However, even in WordNet, among 21,479 adjectives, there are only 1,699 adjectives with <attribute> and/or <see also> pointers. These useful relations seem to be still insufficient in WordNet 3.0. For example, though an <attribute> for "low" is [degree, grade, level], an <attribute> for "cheap" does not exist. We cannot easily trace hypernyms for "cheap" even via the pointer of "low".

In the Japanese WordNet, such relations as <attribute> and <see also> have not been completed yet, and we are trying to detect attribute concepts in Japanese for which adjectives express their values. We are currently proceeding with our investigation on attribute–value relations between nouns and adjectives in Japanese.

There are expressions containing an attribute and its value in a sentence in Japanese.

Examples 3:

a. テーブルが 安い 値段に なった。
teeburu-ga *yasui* *nedan-ni* *natta*
(table) (cheap) (price) (became)
The price of a table became a cheap price.
b. テーブルが 安く なった。
t*eeburu-ga* *yasuku* *natta*
(table) (cheap in adverbial form) (became)
A table became cheap.

In the above examples, the meaning of "安くなる *yasuku naru* (become cheap)" is equal to "安い値段になる *yasui nedan-ni naru* (become a cheap (low) price)", that is, "値段 *nedan* (price)" can be omitted when the adnominal usage of "安い *yasui*

(cheap in adnominal form)" changes to the adverbial form "安く *yasuku* (cheap in adverbial form)". In this case "値段 *nedan* (price)" is a meaning implied by "安く *yasuku in adverbial usage* (cheap)".

5 Future Direction

As shown in the above sections, both the English and Japanese WordNet are not sufficient in the adjective domain, e.g., an attribute concept of "cheap" does not exist.

There are several methods for finding categories of words: [2], [3], [4], [5], [6]. These methods showed good results for generating hypernym concepts mainly from nouns and verbs. We have to develop a method useful for finding relations between attribute concepts and adjectives.

As a future direction we will contribute to the development of the English and Japanese WordNet by considering what are attribute concepts for adjectives.

Acknowledgment. We are grateful for discussion and resources from the Kyoto Project which is co-funded by EU – FP7 ICT Work Programme 2007 under Challenge 4 – Digital libraries and Content, Objective ICT- 2007.4.2 (ICT-2007.4.4): Intelligent Content and Semantics (challenge 4.2).

References

1. Bond, F., Isahara, H., Kanzaki, K., Uchimoto, K.: Boot-strapping a WordNet using multiple existing WordNets. In: 6th International Conference on Language Resources and Evaluation (LREC 2008), pp. 1619–1624 (2008)
2. Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: 37th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 120–126 (1999)
3. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: 14th International Conference on Computational Linguistics (COLING), pp. 539–545 (1992)
4. Lin, D., Pantel, P.: Concept Discovery from Text. In: 19th International Conference on Computational Linguistics (COLING), pp. 768–774 (2002)
5. Pantel, P., Ravichandran, D.: Automatically Labeling Semantic Classes. In: Human Language Technology/North American Chapter of Association for Computational Linguistics (HLT/NAACL), pp. 321–328 (2004)
6. Ryu, P.M., Choi, K.S.: Taxonomy Learning using Terms Specificity and Similarity. In: 2nd Workshop on Ontology Learning and Population, held at COLING/ACL-2006, pp. 41–48 (2006)
7. Vossen, P. (ed.): Euro WordNet. Kluwer, Dordrecht (1998)
8. Vossen, P.: Building wordnets (2005), `http://www.globalwordnet.org/gwa/BuildingWordnets.ppt`
9. Wordnet, `http://wordnet.princeton.edu/`
10. Japanese WordNet, `http://nlpwww.nict.go.jp/wn-ja/index.ja.html`

Comparing SMT Methods for Automatic Generation of Pronunciation Variants

Panagiota Karanasou and Lori Lamel

Spoken Language Processing Group, LIMSI-CNRS
91403 Orsay, France
{pkaran,lamel}@limsi.fr

Abstract. Multiple-pronunciation dictionaries are often used by automatic speech recognition systems in order to account for different speaking styles. In this paper, two methods based on statistical machine translation (SMT) are used to generate multiple pronunciations from the canonical pronunciation of a word. In the first method, a machine translation tool is used to perform phoneme-to-phoneme (p2p) conversion and derive variants from a given canonical pronunciation. The second method is based on a pivot method proposed for the paraphrase extraction task. The two methods are compared under different training conditions which allow single or multiple pronunciations in the training set, and their performance is evaluated in terms of recall and precision measures.

Keywords: P2P conversion, pronunciation lexicon, SMT, Moses, pivot paraphrasing.

1 Introduction

Pronunciation variation is one of the factors that influences the performance of an automatic speech recognition (ASR) system, especially for spontaneous speech. Predicting pronunciation variations, that is, alternative pronunciations observed for a linguistically identical word, is a complicated problem and depends on a number of factors, such as the linguistic origin of the speaker, the speaker's education and socio-economic level, the speaking style and conversational context and the relationship between interlocutors. The construction of a good pronunciation dictionary is thus critical to ensure acceptable ASR performance [8]. Moreover, the number of pronunciation variants that need to be included in a dictionary depends on the system configuration [1].

A variety of methods have been proposed in order to obtain pronunciation variants from the canonical pronunciations (baseforms) of words and can be broadly grouped into data-based and knowledge-based methods. Knowledge-based methods using phonological rules [3], [17], require specific linguistic skills, are not language-independent and do not always capture the irregularities in natural languages. By contrast, the data-driven approaches are based on the idea that given enough examples it should be possible to predict the pronunciation of unseen words (in the grapheme-to-phoneme task) or generate multiple

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 167–178, 2010.

pronunciations for improved speech recognition. In this paper, the latter task of generation of pronunciation variations is addressed using data-based methods. Other data-based methods proposed in the literature for the modeling of pronunciation variations include the use of neural networks [4], confusion tables [14] and decision trees [16] or automatic generation of rules for phoneme-to-phoneme conversion [6]. All these methods predict a phoneme for each input symbol using the input symbol and its context as features, but ignore any structure in the output. The methods proposed in this paper take advantage of both the input and the output context and can predict variable length phoneme sequences.

The two methods presented in this paper aim to automatically generate pronunciation variants of words for which a canonical pronunciation is available. The first method is based on the simple use of Moses [7], a publicly-available phrase-based statistical machine translation tool, as a phoneme-to-phoneme converter to generate an n-best list of pronunciation variants. The second method is based on a paraphrase method that uses bilingual parallel corpora and is founded on the idea that paraphrases in one language can be identified using a phrase in another language as a pivot. In the case of multiple pronunciation generation, sequences of modified phonemes found in pronunciation variants are identified using a sequence of graphemes in the corresponding word as a pivot.

The paper is organized as follows. Section 2 describes the two methods used in this study. Section 3 describes the experimental framework and details about the corpora used and the training conditions applied. Section 4 presents the evaluation results of the automatic generation of multiple pronunciations in terms of precision and recall. Conclusions and some discussions for future work are reported in Section 5.

2 Phoneme-to-Phoneme Conversion

This section presents the two proposed methods in detail and compares their strengths and weaknesses, pointing out their utility in different situations. Both methods aim to produce pronunciation variants of the initial (canonical) phonemic transcription. Since the canonical pronunciations are not explicitly indicated in the master lexicon (see Section 3.1), the longest one is taken as the canonical form since the reduced forms often correspond to variants found in conversational speech. The first method generates pronunciation variants using only the phonemic transcriptions of words, while the second method makes use of both the orthographic and phonemic transcriptions and thereby permits to the system to also benefit from the information provided by the orthographic transcription of a word. As we will see later in the results analysis, this last characteristic of the second method is particularly useful under certain training conditions.

2.1 Moses as a Phoneme-to-Phoneme Converter

Moses has already been proposed for the grapheme-to-phoneme (g2p) conversion [5], [9] task. A pronunciation dictionary is used in the place of an aligned bilingual text corpora. The orthographic transcription is considered as the source

language and the pronunciation as the target language. In the case that the pronunciation dictionary has a reasonable coverage of the language of interest, this method can be successfully used for g2p conversion because it has all the desired properties of a g2p system. To predict a phoneme from a grapheme, it takes into account the local context of the input word from a phrase-based model and allows sub-strings of graphemes to generate phonemes. The phoneme sequence information is modeled by a phoneme n-gram language model that corresponds to the target language model in machine translation. More technical details on the Moses components will be given in the Section 3.2. These properties are also desired for a p2p converter, which has, moreover, higher potential for capturing pronunciation variation phenomena in languages like English, where orthography and pronunciation generally have a looser relationship than in other languages. A second direction explored in this work is based on the idea of seeing the use of SMT tools with a monolingual corpora for paraphrase generation [11] as being analogous to generating pronunciation variants. These similar approaches to two distinct problems led us to the idea of trying to use Moses for p2p conversion. In this case, the source language and the target language are aligned phonemic transcriptions. As the source language we define the canonical pronunciation (the longest one[1]) and as target language itself and/or its variants depending on their existence or not in the different versions of the training set as presented in the Section 3.1.

2.2 Pivot Paraphrasing Approach

This method is based on the one presented in [2]. Paraphrases are alternative ways of conveying the same information. We can easily see the analogy with multiple pronunciations of the same word. The multiple pronunciations are alternative phonemic expressions of the same orthographic information.

In [2], a bilingual parallel corpus is used to show how paraphrases in one language can be identified using a phrase in another language as a pivot. In the problem of automatic generation of pronunciation variants, a corpus of word-pronunciation pairs is used as the analogy of the aligned bilingual corpus, instead of the pronunciation-pronunciation aligned corpus in the previous method. The idea is to define a paraphrase (pronunciation variant) probability that allows paraphrases (pronunciation variant sequences) extracted from a bilingual parallel corpus to be ranked using translation probabilities, and then rerank the generated pronunciation variants taking the contextual information into account. The translation table that is used is extracted by Moses. In [2], the authors look at the English translation of foreign language phrases, find all occurrences of those foreign phrases, and then look back to determine to what other English phrases they correspond. The other English phrases are seen as potential paraphrases. In the pronunciation generation case, we look at all the entries in the translation table, find the sequences of graphemes to which a sequence of phonemes is translated, and then look back to what other sequences of phonemes the particular sequence of graphemes is translated.

[1] Most of the variants reflect reduced pronunciations found in casual speech.

Phrase alignments in a parallel corpus are used as pivots between English (pronunciation) paraphrases. These two-way alignments are found using recent phrase-based approaches to statistical machine translation. In the following definitions, f is a graphemic sequence and e_1 and e_2 are phonemic sequences. The paraphrase probability $p(e_2 \mid e_1)$ is assigned in terms of the translation model probabilities $p(f \mid e_1)$ and $p(e_2 \mid f)$. Since $e1$ can be translated as multiple foreign language phrases (graphemic sequences), we sum over f:

$$\hat{e}_2 = \arg\max_{e_2 \neq e_1} p(e_2 \mid e_1) \tag{1}$$

$$= \arg\max_{e_2 \neq e_1} \sum_f p(f \mid e_1) p(e_2 \mid f) \tag{2}$$

This returns the single best paraphrase, $\hat{e}_2$, irrespective of the context in which e_1 appears. Since, the best paraphrase may depend on information about the sentence that e_1 appears in, the paraphrase probability can be extended to include the sentence S:

$$\hat{e}_2 = \arg\max_{e_2 \neq e_1} p(e_2 \mid e_1, S) \tag{3}$$

This allows the candidate paraphrases to be ranked based on additional contextual information in the sentence S. A simple language model probability is included, which can additionally rank e_2 based on the probability of the sentence formed by substituting e_2 for e_1 in S. The language model is trained on the correct pronunciations of the training set. For the reranking based on the language model, we use the SRI toolkit [13]. Finally, some more pruning is done on the reranked list keeping the maximum of ten, five or one pronunciation variants per canonical pronunciation without changing the order of the elements of the reranked list.

An example of a paraphrase pattern in the pronunciation dictionary is[2]:

discou**nt**ed dIskW**nt**xd
discou**nt**ed dIskW**n**xd
discou**nt**enance dIskW**n**Nxns
discou**nt**enance dIskW**nt**Nxns

The alternative pronunciations differ only in the part that can be realized as either **nt** or **n**, while the rest remains the same.

3 Experiments

3.1 Corpus

The LIMSI Master American English dictionary serves as basis of this work. It is a pronunciation dictionary with 187975 word entries (excluding words starting with numbers) with on average 1.2 pronunciations per word. The pronunciations are represented using a set of 45 phones [8], each phone corresponding to a

[2] The phone set used is given in Table 1 of [8].

single character. The dictionary has been created with extensive manual supervision. Each dictionary entry has the orthographic transcription of a word and its pronunciations (one or more). 18% of the words are associated with multiple pronunciations. The majority of words have only one pronunciation, leaving it to the acoustic model to represent the observed variants in the training set that are due to allophonic differences. Moreover, since the dictionary is mostly manually constructed, it is certainly incomplete with respect to coverage of pronunciation variants particularly for uncommon words. The pronunciations of words of foreign origin (mostly proper names) may also be incomplete since their pronunciation depends highly on the speaker's knowledge of the language of origin. This means that some of the automatically generated variants are likely to be correct (or plausible) even if they are not in the current version of the Master dictionary.

Case distinction is eliminated since in general it does not influence the word's pronunciation, the main exceptions being the few acronyms which have a spoken and spelled form. Some symbols in the graphemic form are not pronounced, such as the hyphen in compound words. The dictionary contains a mix of common words, acronyms and proper names. It should be noted that these last categories are difficult cases for g2p or p2p converters and particular effort has been made to pronounce proper names in text-to-speech synthesis technology [12].

The corpus is randomly split into a training, a development (dev) and a test set. The dev set is necessary for the optimisation of the weights of Moses model as will be later explained (tuning) and the test set is used for the evaluation of the system. This division is based on dictionary entries so that all the pronunciations of a given word will be in the same set. If not, we would have the paradox of training the system with certain pronunciations and asking it to generate only the different pronunciations of the same word found in the test set.

The dev set has 9000 entries and the test set 16000 entries. The original dictionary entries of training, dev and test sets were transformed to have one graphemic transcription-pronunciation pair per entry as opposed to one entry corresponding to the graphemic transcription of a word with all its pronunciation variants. This is to have a format that resembles the aligned parallel texts used for training machine translation models. After transformation, the dev and test sets have 11196 and 19782 distinct entries. All the results are calculated for the same test set, so that their comparison is legitimate. Three different training conditions are compared for the two p2p systems:

1. Train on the entire dictionary. Words may have one or more pronunciations (tr_set).
2. Train only on words with two or more pronunciation variants. All words have multiple pronunciations (tr_set_m).
3. Train on the entire dictionary using only the longest (canonical) pronunciation to have one pronunciation per word (tr_set_l).

At this point, a further preparation of the training set for each method is required. For the method where Moses is used as a p2p converter, a "monolingual"

parallel corpus is needed, meaning that both the source language and the target language will have phonemes as elements. The source language is always formed by the canonical pronunciation segmented into phonemes. The target language is formed of the corresponding pronunciations depending on the training condition. For the pivot method, the training set is used as a parallel corpus with one graphemic transcription-pronunciation pair per line with spaces separating characters, in order to use Moses (as in a g2p task) to generate a translation table that will be used to extract paraphrased sequences. Each word is a source sentence with each grapheme being an element of the source sentence and each pronunciation is a target sentence with each phoneme forming an element of the target sentence.

Table 1 gives an overview of the data sets used with the number of entries (distinct pairs) and the average number of pronunciations/word in the three training conditions after preprocessing.

Table 1. Training conditions

Training set	Number of entries	Average number prons/word
tr_set	201423	1.2
tr_set_m	67769	2.3
tr_set_l	162974	1.0

It can be seen in Table 1 that there are large differences for the three training conditions. For tr_set_m the number of entries diminishes to one third of the original dictionary. However, the number of pronunciations per word almost doubles. In this case, the extra information given by the canonical pronunciations of words with only one pronunciation is lost, but we allow the systems to change the frequency relationship between the phrases of the canonical pronunciations and the phrases found in pronunciation variants, and see how this influences the generation of pronunciation variants which is our main interest in this work. In the third training condition, only the canonical pronunciation of each word is kept in the training data. This allows us to see if pronunciation variants can be generated even under limited training conditions. For example, this condition corresponds to generating variants from the output of a rule-based g2p system which, if originally developed for speech synthesis, may not model pronunciation variants or to enriching a dictionary with limited pronunciation variants.

3.2 System

The system that is used to train the models in both methods is based on Moses. In the first proposed method, Moses is used as a p2p converter in an one-stage procedure. Besides the phrase (translation) table, a phoneme-based 5-gram language model is built on the pronunciations in the training set using the SRI toolkit [13]. Moses also calculates a distortion model, but our dictionary does not include a sufficient number of metathesis cases, so the monotonic decoding

does not change the final results. Finally, the combination of all components is fully optimized with a minimum error training step (tuning) on the dev set. The tuning strategy we used was the standard Moses training framework based on the maximization of the BLEU score [10]. The optimized weights generated by tuning are added to the configuration file. Moses can also provide an n-best translation list. This list gives the n-best translations of a source string with the distortion, the translation and the language model weights, as well as an overall score for each translation. As stated earlier we keep only the 1-, 5- or 10-best translations (i.e. pronunciation variants) per canonical pronunciation. Some pronunciations have fewer possible variants, in which case all variants are taken.

In the pivot method, generating pronunciation variants is a four-stage procedure. Moses is used in the first stage for g2p conversion and extraction of the translation table. In the second stage, the paraphrased pairs with their probabilities are extracted from the canonical pronunciations of the test set as previously described. The 10-best paraphrases for each input phonemic sequence are extracted with a maximum length of 3 for the extracted paraphrases. In the third stage, the paraphrases are substituted in the canonical pronunciations of the test set for all their occurrences with all the possible combinations (only in the first occurrence, only in the second occurrence, in the first and in the second occurrence, etc.), limiting to 3 the maximum number of occurrences of the same paraphrase in a canonical pronunciation. In the fourth and final stage the generated list of pronunciation variants is reranked based on the context. The context is expressed by the same phoneme-based 5-gram language model used in the first method. The SRI toolkit is used to rerank the multiple pronunciation n-best list modifying its probabilities. As for the first method, the 1-, 5- or 10-best variants are kept for each canonical pronunciation in the ordered list.

4 Evaluation

Different measures have been proposed to evaluate the predictions of pronunciation variants derived from the original "canonical" form. The most frequently used are precision and recall, first introduced in information retrieval [15]. The canonical pronunciations x_i of the test set can have one or more variants y_i (y_i is a set). Moreover, our systems can generate one or more variants $f(x_i)$ ($f(x_i)$ is also a set). Thus, the recall that corresponds to a couple $(y_i, f(x_i))$ is the number of correct generated variants for a canonical pronunciation in the test set divided by the number of correct variants given in the test set for this canonical pronunciation:

$$r_i = \frac{|f(x_i) \cap y_i|}{|y_i|} \tag{4}$$

The precision is the number of correct generated variants divided by the number of generated variants:

$$p_i = \frac{|f(x_i) \cap y_i|}{|f(x_i)|} \tag{5}$$

The total recall is the mean value of the recall of each example:

$$r = \frac{1}{n} \sum_{i=1}^{n} r_i \tag{6}$$

Analogously, the total precision is the mean value of the precision of each example. We refer to the previous definitions as micro-recall and micro-precision respectively. If the examples are normalized by the number of expected variants (correct variants in the reference), the total recall becomes:

$$r = \frac{\sum_{i=1}^{n} |r_i||y_i|}{\sum_{i=1}^{n} |y_i|} \tag{7}$$

In this last case, the macro-recall is defined. Macro-precision is defined analogously. The macro-measures give more weight to the examples with multiple variants, while the micro-measures consider all the examples equally weighted.

It is important to do the evaluation on a pair level and not just consider the error rate of generated pronunciations, to avoid counting as correct a generated pronunciation that does not correspond to the canonical pronunciation it is associated with in the reference. There is always the possibility that our system will generate a pronunciation out of a baseform (i.e. canonical pronunciation) that is not a variant of this baseform, but, however, is a correct variant of another baseform. This is counted as a false generation. Another thing that should be noted is that we prune the canonical pronunciation-pronunciation variant pairs that do not include a new variant. This is important in order to improve the precision because while the pivot method generates only pronunciation variants, when Moses is used as p2p converter it often outputs the canonical pronunciation that was used as input because it learns from the training data that the most probable pronunciation corresponding to a given canonical pronunciation is usually itself. This depends a lot on the training conditions and makes this method inappropriate in certain cases.

To control the precision of our systems, an upper limit is put to the number of n-best variants that are kept in the hypotheses. The 1-, 5- and 10-best variants per canonical pronunciation are generated consecutively. The n-best list is limited to 10 because preliminary studies showed that larger n only slightly improves recall while severely degrades precision. There is quite a bit of overgeneration, since in the 19k pronunciation-pronunciation pairs of the test set there are only 4k pairs with pronunciation variants. This could not be avoided with a random selection of the test set from the original dictionary where only 18% of words have variants as already stated. However, there is the possibility that some of the generated variants which are not in the reference (and therefore counted as errors) could be considered acceptable by a human judge. Evaluating the system by an automatic measure cannot take into account the potential lack of coverage of the reference dictionary.

The two systems, Moses as phoneme-to-phoneme converter (m_p2p) and the pivot paraphrasing method (p_p2p) were tested for the 3 training conditions presented earlier. The results using the two proposed evaluation metrics are

Table 2. Results using Moses as phoneme-to-phoneme converter for the 3 training conditions

Training set	Measure	1-best	5-best	10-best
tr_set	Micro-recall	0.20	0.75	0.83
	Macro-recall	0.19	0.73	0.81
tr_set_m	Micro-recall	0.21	0.75	0.80
	Macro-recall	0.19	0.74	0.80
tr_set_l	Micro-recall	–	0	0
	Macro-recall	–	0	0

Table 3. Results using the pivot paraphrasing method for the 3 training conditions

Training set	Measure	1-best	5-best	10-best
tr_set	Micro-recall	0.29	0.60	0.70
	Macro-recall	0.26	0.56	0.66
tr_set_m	Micro-recall	0.25	0.56	0.70
	Macro-recall	0.22	0.53	0.66
tr_set_l	Micro-recall	0.09	0.26	0.38
	Macro-recall	0.09	0.24	0.35

shown in Tables 2 and 3 respectively. We only present recall measures in the tables because this is what is of most interest in the particular task. It is more important to cover possible pronunciations than to have too many since other methods can be applied to reduce the overgeneration (alignment with audio, manual selection, use of pronunciation probabilities, etc). The best value that both precision and recall can obtain is 1. However, the best value of precision is often further limited depending upon the number of elements of the n-best list and the overgeneration that cannot be avoided.

As can be expected, for both methods the number of correctly generated variants increases with the size of the n-best list. This is normal not only because the number of hypothesis increases with the size of the n-best list, but also because there are canonical pronunciations in the test set that have more than one variant (approximately 1/6 of the part of the test set with multiple pronunciations) which cannot be captured when an insufficient number of pronunciations is generated.

It is also interesting to compare the results of the first and the second training conditions (whole dictionary vs. keeping only entries with multiple variants) for the two methods. In the second case, the amount of training data is only one third of the original training set. However, the results are more or less the same for both training conditions. This may be because the information that the model is using to learn how to generate variants is mostly captured by the multiple pronunciations in the training set and less by the fewer variations observed in the canonical pronunciations of one-pronunciation words. What the model is learning in this case is focused on the relationship between the canonical pronunciation and other variants, and therefore has effectively more relevant information and it does not get watered down by the self-production. This may compensate for the reduced amount of data.

A comparative analysis of the two methods can also be made. In the first (whole training set) and in the second (only entries with variants) training conditions, using Moses as a p2p converter gives better results in terms of the generation of pronunciation variants for both micro and macro measures when the 5-best and the 10-best variants are kept. However, when only the 1-best generated pronunciation is kept, the pivot method gives better results. This is due to the generation of canonical pronunciations by Moses when used as p2p converter, which are subsequently removed from the results because they already exist in the input. The number of variants generated by Moses-p2p when only the 1-best is kept are quite limited. This is why, while the recall is lower than that of the pivot method, the precision is higher.

It can be seen that the results change when the training set is limited to the canonical pronunciations only (the third condition). In this case the pivot method manages to produce some results, while for Moses the model fails to generate any variants (this is why the corresponding columns are left empty in Table 2) and all the variants that are in the 5-best or the 10-best lists are false. These results warrant a bit more discussion. The results are promising for the pivot method, because even when the training dictionary has few or no pronunciation variants, the pivot method can still be used to generate some alternative pronunciations. This can be explained by the fact that the pivot method uses also the graphemic information. Even if no variants are included in the training set, it can still find graphemic sequences of words that correspond to different phonemic sequences and consider these phonemic sequences as possible modifications of pronunciations. For example, in the training set the word "autor**out**e" is pronounced "ctor**ut**" and the word "sh**out**ing" is pronounced "s**Wt**IG". These words have the graphemic sequence "**out**" in common which can be used as a pivot between the phonemic sequences "**ut**" and "**Wt**". These phonemic sequences become a paraphrased pair that generates correctly the variants "r**Wt**s" and "r**ut**s" of the word "r**out**es" found in the test set.

This illustrates the difficulty of generating pronunciations in English, because the correspondence between orthographic forms and canonical pronunciations does not follow strict rules which would prevent the pivot method from finding modified phonemic sequences corresponding to the same graphemic sequence. This is not the case when Moses is used as phoneme-to-phoneme converter. When no variants are given to the system, it does not have any additional information in order to be trained for the task of generating multiple pronunciations. It is like trying to train an SMT system without a target language. It can just learn to align the phonemic sequences with themselves, which is fine for the g2p task, but is not applicable to the generation of variants. In this case, is it wrong to use Moses for this task as it is obvious that it has nothing to learn from the training data.

5 Conclusions

This paper has reported on applying two data-based approaches inspired from research in statistical machine translation to the problem of generating pronunciation variants. One of the objectives of this work was to compare these

two approaches to modeling pronunciation variations. The approaches differ in the way that information about pronunciation variation is obtained. The approach using Moses as phoneme-to-phoneme converter takes into account only the information provided by the phonemic transcriptions. The pivot method uses information from both the phonemic and the orthographic transcriptions.

When the full dictionary (that contains words with one or more pronunciations) is used for training, the Moses-based method gives better results than the pivot-based one. This is also the case when training is carried out on only entries with multiple pronunciations. However, when the training dictionary does not contain any pronunciation variants, the Moses-based method cannot be used, while pivot can still learn to generate variants. This is an advantage of the pivot method, and could be useful for languages without well-developed multiple-pronunciation dictionaries. This arises from the use of information provided by the orthographic transcription by the pivot method. An interesting follow-up study is to use the pivot method to propose variants as a post-processing step to a g2p system. Another case to study is the influence of the p2p converter on the results of a g2p converter if their output n-best lists are combined. We have started some preliminary experiments in these directions with promising results.

In future work we will evaluate the proposed methods at generating pronunciations and variants for proper names, which are the most difficult cases to handle. These also account for the majority of words that need to be added to a dictionary once a reasonably sized one is available for the given language.

Another important outstanding issue concerns the proper way to evaluate the ability of a system to generate pronunciation variants. In this work, recall and precision have been used, however other measures such as phoneme accuracy can also be applied. In this case it may be appropriate to have phone-class dependent penalties with certain confusions being more important than others. In order to improve the precision, the n-best lists need to be more heavily pruned. One direction to explore is using audio data to remove pronunciations, however this can only apply to words found in the audio data.

The ultimate test of course is how the variants affect the accuracy of a speech-to-text transcription system.

Acknowledgments. This work is partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation and by the ANR EdyLex project.

References

1. Adda-Decker, M., Lamel, L.: Pronunciation variants across system configuration, language and speaking style. Speech Communication 29, 83–98 (1999)
2. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proc. of ACL, pp. 597–604 (2005)
3. Divay, M., Vitale, A.-J.: Algorithms for grapheme-phoneme translation for English and French: Applications for database searches and speech synthesis. Computational linguistics 23(4), 495–523 (1997)

4. Fukada, T., Yoshimura, T., Sagisaka, Y.: Automatic generation of multiple pronunciations based on neural networks. Speech Communication 27(1), 63–73 (1999)
5. Gerosa, M., Federico, M.: Coping with out-of-vocabulary words: open versus huge vocabulary ASR. In: ICASSP, pp. 4313–4316 (2009)
6. Heuvel, H., van de Reveil B., Martens, J.-P.: Pronunciation-based ASR for names. In: Proc. of Interspeech, pp. 2991–2994 (2009)
7. Koehn, P., et al.: Moses: Open source toolkit for statistical machine translation. In: Proc. of ACL (2007)
8. Lamel, L., Adda, G.: On designing pronunciation lexicons for large vocabulary, continuous speech recognition. In: Proc. ICSLP 1996, pp. 6–9 (1996)
9. Laurent, A., Deleglise, P., Meignier, S.: Grapheme to phoneme conversion using a SMT system. In: Proc. of Interspeech (2009)
10. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proc. of ACL, pp. 311–318 (2002)
11. Quirk, C., Brockett, C., Dolan, W.: Monolingual Machine Translation for Paraphrase Generation. In: Proc. of EMNLP, pp. 142–149 (2004)
12. Spiegel, M.F.: Using the ORATOR synthesizer for a public reverse-directory service: design, lessons, and recommendations. In: EUROSPEECH 1993, pp. 1897–1900 (1993)
13. Stolcke, A.: SRILM-An extensible language modeling toolkit. In: Proc. ICSLP 2002, vol. 2, pp. 901–904 (2002)
14. Tsai, M.-Y., Chou, F.-C., Lee, L.-S.: Pronunciation modeling with reduced confusion for mandarin chinese using a three-stage framework. IEEE Transactions on Audio, Speech and Language Processing 15(2), 661–675 (2007)
15. Van Rijsbergen, C.J.: Information Retrieval. Butterworths, London (1979)
16. Weintraub, M., Fosler, E., Galles, C., Kao, Y.-H., Khudanpur, S., Saraclar, M., Wegmann, S.: WS96 project report:Automatic learning of word pronunciation from data. In: JHU Workshop Pronunciation Group (1996)
17. Wester, M.: Pronunciation modeling for ASR- Knowledge-based and data-driven methods. Comput. Speech Lang., 69–85 (2003)

Automatic Learning of Discourse Relations in Swedish Using Cue Phrases

Stefan Karlsson and Pierre Nugues

Lund University
Lund Institute of Technology
Department of Computer Science
Box 118
S-221 00 Lund, Sweden
stefan.karlsson.342@student.lth.se, Pierre.Nugues@cs.lth.se

Abstract. This paper describes experiments to extract discourse relations holding between two text spans in Swedish. We considered three relation types: cause-explanation-evidence (CEV), contrast, and elaboration and we extracted word pairs eliciting these relations. We determined a list of Swedish cue phrases marking explicitly the relations and we learned the word pairs automatically from a corpus of 60 million words. We evaluated the method by building two-way classifiers and we obtained the results: Contrast vs. Other 67.9%, CEV vs. Other 57.7%, and Elaboration vs. Other 52.2%.

The conclusion is that this technique, possibly with improvements or modifications, seems usable to capture discourse relations in Swedish.

Keywords: rhetorical relations, discourse relations, cue phrases, naïve Bayes classification.

1 Introduction

Rhetorical relations and the *Rhetorical structure theory* [1] form a framework to describe and interpret the organization of a text. In this theory, relations consist of annotated links tying two text spans as, for example, the clauses in the sentence:

> Malaria förekommer framför allt i sumpiga trakter, därför att myggglarverna utvecklas väsentligen i stillastående vattensamlingar.
> "Malaria exists primarily in wetlands, because the mosquito larvae develops in still waters."
> [2, Uggleupplagan, vol. 8:1490].

The next sentence gives another example of a rhetorical relation between two clauses:

> Till en början utgafs tidningen en gång i veckan, men i dec. 1850 förvandlades den till daglig

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 179–184, 2010.

"Initially the newspaper was published once a week, but in Dec. 1850 it was transformed into a daily"
[2, Uggleupplagan, vol. 3:1157].

Rhetorical relations can be associated with certain cue words or phrases, such as *därför* 'because' with explanations in the first example and *men* 'but' with contrasts in the second one. Nonetheless, cue phrases are often ambiguous. If you change *but* to *and* in the second example, a reader would probably conclude that the relation tying the two spans remains the same. However, since *and* is not a discourse marker as explicit as *but*, it cannot be used in a one-to-one association to identify a relation. A more elaborate strategy is then necessary to extract and label rhetorical relations.

First techniques to automatically identify different types of discourse relations used discourse markers and were based on manually-written rules as in [3] and [4]. Most algorithms described in the literature have only been applied to English or Japanese.

This paper describes a system that decides whether two text spans in Swedish can be classified as being tied by a particular discourse relation. In this system, we implemented and adapted Marcu and Echihabi's algorithm [5], which automatically learns relations from minimally annotated texts. A useful application of the analysis of rhetorical relations would be to extract all causes of a fact and put them into a knowledge base.

2 A Statistical Model

Some word pairs are frequent in contrasts, hypothetically for example, *week* and *daily*, as in the example above, and other pairs in explanations, i.e. *exists* and *develops*. Instead of extracting relations with manual rules, we can try to derive automatically sets of words involved in specific relations from corpora.

Marcu and Echihabi proposed an unsupervised method [5] to train naïve Bayesian classifiers based on this idea. The first step extracts contiguous text spans using a set of predefined markers and forms the Cartesian product of the words in them. Let W_1 and W_2 be two contiguous text spans. The second step counts all the word pairs $(w_i, w_j) \in W_1 \times W_2$ of the contiguous text spans extracted from the corpus.

The probability that two text spans are tied by a particular relation is calculated as follows:

$$P(r_k|W_1, W_2) = \frac{P(W_1, W_2|r_k)P(r_k)}{P(W_1, W_2)}. \quad (1)$$

Using the naïve Bayes strategy, we estimate $P(W_1, W_2|r_k)$ as $\prod P((w_i, w_j)|r_k)$, where w_i and w_j stand for the words in each span.

3 Experimental Setup

3.1 Extraction of Text Spans

We considered three discourse relations: *cause-explanation-evidence* (CEV), *contrast*, and *elaboration*. We compiled a Swedish corpus using texts from the Runeberg project (45 million words) and the European Parliament proceedings [6] (16 million words), a multilingual corpus, where we used the Swedish source parts. We then inspected the corpus manually and incrementally built the extraction patterns shown in Table 1.

Table 1. Swedish extraction patterns used in the experiments. BOS indicates the beginning of the sentence and EOS, the end of the sentence.

Contrast
[BOS ...][, men ... EOS]
[BOS ...][, ehuru ... EOS]
[BOS ...][, fastän ... EOS]
[BOS ...][, trots att ... EOS]
Cause-Explanation-Evidence
[BOS ...][, därför att ... EOS]
[BOS ...][, eftersom ... EOS]
[BOS ... EOS][BOS Alltså ... EOS]
[BOS ...][, alltså ... EOS]
[BOS ... EOS][BOS Således ... EOS]
[BOS ...][, således ... EOS]
[BOS ... EOS][BOS Sålunda ... EOS]
[BOS ...][, sålunda ... EOS]
[BOS ...][, ty ... EOS]
[BOS ... EOS][BOS Ty ... EOS]
[BOS ... EOS][BOS Därför ... EOS]
Elaboration
[BOS ...][vilket ... EOS]
[BOS ...][hvilket ... EOS]

The *Nordisk Familjebok* encyclopedia from the end of the 19th century and the beginning of the 20th century represents a large part of the corpus. This explains why we had to use words like *ty* 'because' and *ehuru* 'in spite of' as markers that do not belong to present day Swedish.

The corpus was randomly divided into a training set (90%) and a test set (10%). To improve training, we used only verbs and nouns [5]. We tagged the corpus words with their part of speech using the Granska tagger [7]. We kept the nouns and the verbs and discarded the rest of the words, including the markers from the patterns.

Finally, we compiled the training examples: 130,796 contrasts, 37,319 CEV, and 43,387 elaborations, and a test set of 14,643 contrasts, 4,107 CEV, and 4,976 elaborations, all extracted using the patterns in Table 1.

3.2 Evaluation Methods

For the evaluation, we built binary classifiers to distinguish:

- Contrast vs. Other,
- CEV vs. Other, and
- Elaboration vs. Other,

where Other stands for an equal amount of relations of the other two types as CEV+Elaboration in the first case. A decision is made by taking the maximum of $P(r_k|W_1, W_2)$ for each relation. In Equation 1, $P(W_1, W2)$ can be discarded, since it is the same in all the relations.

In the evaluation, we build sets of equal proportions to eliminate the factor $P(r_k)$. In the Contrast vs. Other case, we extracted 8,000 contrasts, 4,000 CEVs, and 4,000 elaborations from the test set. In the CEV vs. Other case, we used 4,000 CEVs, 2,000 contrasts, and 2,000 elaborations. Finally in the Elaboration vs. Other case, we used 4,000 elaborations, 2,000 contrasts, and 2,000 CEVs.

We found that the Laplace method shifted too much mass of probability to unseen word pairs. Therefore, we used Lidstone's rule instead, which amounts to setting [8]:

$$P((w_1, w_2)|r_k) = \frac{(count + \lambda)}{(total + \lambda \cdot cardinal)}, \tag{2}$$

where *cardinal* is the number of entries in the table. We found that a lambda of 0.05 seemed to maximize the accuracy of the classifiers. In a similar experiment, [9] used the value of 0.25.

3.3 Results

Table 2 shows the accuracy of the classifiers. A result of 67.9% in the Contrast vs. Other condition is in the same range as the results obtained for English [5], which reported between 60% and 70% for most relations. The results for Elaboration vs. Other that reached 52.2% were significantly lower, however.

Table 2. The accuracy of each classifier. In each case, the baseline is 50%.

Relation	Accuracy
Contrast vs. Other	67.9%
CEV vs. Other	57.7%
Elaboration vs. Other	52.2%

4 Conclusions

Results around 60% clearly indicates that the classifier is better than a random assignment of text spans to each class. The result with elaboration is not completely satisfying though, which first of all can be accounted to the fact that we only used 43,387 training examples.

As perspectives, some simple improvements could be made. Since there is no intrinsic order in contrast relations, the table could be made commutative. However, we did not consider it a critical point, since there were more than 130,000 training examples of contrasts. The most critical point though is to find the best set of cues phrases for each discourse relation. The corpora used in this experiment was quite small for the task and we had to use many cue phrases at one time. With a larger training set, we could determine which phrases contribute most to the model without introducing noise; for example by comparing results obtained by including or excluding training examples from a particular extraction pattern.

Not only the size of the corpora limits the performance of this technique. The example words that indicate a contrast, i.e. *week* and *daily* in the example in Sect. 1 can possibly stand in other types of discourse relations. Such overlapping word pairs will dim the statistical accuracy of the model no matter the size of the corpora. This is a major limitation of the general approach taken and can only be dealt with by introducing other types of classification information to distinguish between the rhetorical relations. In English, possibly WordNet [10,11] or FrameNet [12] could be used to figure out which word pairs indicate a particular relation.

To sum up, we presented evidence of a feasible technique for the automatic extraction of discourse relations in Swedish. Marcu and Echihabi showed [5] that using this technique as a complement to extracting cue-phrase marked sentences, can increase the number of correctly classified contrasts from 26% to 77%. Further investigations are however necessary to evaluate more accurately the applicability of this algorithm in Swedish.

Acknowledgments. Lars Aronsson provided most of the corpora used from the Runeberg project. Jonas Sjöbergh at Kungliga Tekniska Högskolan provided the Granska grammatical tool that we used to identify nouns and verbs. Finally, the Granska tool was trained on the Stockholm-Umeå corpus [13].

References

1. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: A theory of text organization. Technical Report RS-87-190, Information Sciences Institute (1987)
2. Meijer, B. (ed.): Nordisk familjebok. Uggleupplagan edn. Nordisk familjeboks förlags aktiebolag, Stockholm (1904–1926)
3. Kurohashi, S., Nagao, M.: Automatic detection of discourse structure by checking surface information in sentences. In: Proceedings of the 15th International Conference on Computational Linguistics, COLING 1994, Kyoto, vol. 2, pp. 1123–1127 (1994)

4. Corston-Oliver, S.: Computing Representations of the Structure of Written Discourse. PhD thesis, University of California, Santa Barbara (1998)
5. Marcu, D., Echihabi, A.: An unsupervised approach to recognizing discourse relations. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics ACL 2002, Philadelphia, pp. 368–375 (2002)
6. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: Proceedings of The Tenth Machine Translation Summit, Phuket, Thailand (2005)
7. Carlberger, J., Kann, V.: Implementing an efficient part-of-speech tagger. Software Practice and Experience 29, 815–832 (1999)
8. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
9. Blair-Goldensohn, S., McKeown, K.R., Rambow, O.C.: Building and refining rhetorical-semantic relation models. In: Proceedings of NAACL HLT 2007, Rochester, NY, pp. 428–435 (2007)
10. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM 38, 39–41 (1995)
11. Fellbaum, C.: WordNet: A Lexical Database for English. MIT Press, Cambridge (1998)
12. Ruppenhofer, J., Baker, C.F., Fillmore, C.J.: The FrameNet database and software tools. In: Braasch, A., Povlsen, C. (eds.) Proceedings of the Tenth Euralex International Congress, Copenhagen, Denmark, vol. 1, pp. 371–375 (2002)
13. Ejerhed, E., Källgren, G., Wennstedt, O., Åström, M.: The linguistic annotation system of the Stockholm-Umeå project. Technical report, University of Umeå, Department of General Linguistics (1992)

The Representation of Diatheses in the Valency Lexicon of Czech Verbs*

Václava Kettnerová and Markéta Lopatková

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

Abstract. In the present paper, we deal with diatheses in Czech from a lexicographic point of view. We propose a method of their description in the valency lexicon of Czech verbs VALLEX. We distinguish grammatical and semantic diatheses as two typologically different changes in verbal valency structure. In case of grammatical diatheses, these changes are regular enough to be described by formal syntactic rules. In contrast, the changes in valency structure of verbs associated with semantic diatheses vary even within one type of diathesis. Thus for the latter type, we propose to set separate valency frames corresponding to their members and to capture the changes in verbal valency structure by lexical rules based on an adequate lexical-semantic representation of verb meaning.

Keywords: valency lexicon, grammatical and semantic diatheses, changes in valency structure of verbs, lexical-semantic representation of verbs.

1 Introduction

Valency behavior of verbs is so heterogenous that it cannot be described by general syntactic rules. Instead, it must be captured in the form of lexical entries separately for each verb. Prototypically, a single meaning of a verb corresponds to a single valency frame. However, in many cases, semantically close uses of verbs can be syntactically structured in different ways. See the following examples:

(1) a. *Peter smeared butter on bread.* – b. *Peter smeared bread with butter.*

(2) a. *Butter was smeared on bread (by Peter).* – b. *Bread was smeared with butter (by Peter).*

The uses of the verb *to smear* illustrated by the examples in (1) and (2) correspond to four different syntactic structures despite their obvious semantic similarity. However, listing separate valency frames for each of them makes the lexicon bigger than expected. Moreover, such a massive polysemy of verbs seems

* The research reported in this paper was carried out under the project of MŠMT ČR No. MSM002162083. It was supported by the grant No. LC536 and partially by the grant No. GA P406/2010/0875.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 185–196, 2010.

to be contraintuitive. Thus we propose to describe the changes in the valency structure of semantically related uses of verbs by means of formal syntactic and lexical rules.

First, let us focus on the pairs in (1a)-(2a) and (1b)-(2b), respectively. The changes in the valency structure of the verb *to smear* are expressed by grammatical means; we refer to the relation between these uses of the verb as a grammatical diathesis. In contrast, the changes in the valency structure of the verb *to smear* in pairs (1a)-(1b) and (2a)-(2b), respectively, are expressed by lexical-semantic means. We refer to the relation between such uses of verbs as a semantic diathesis.

The representation of grammatical and semantic diatheses is proposed here for the valency lexicon VALLEX, which aims at the explicit description of valency behavior of Czech verbs.[1] This lexicon takes the Functional Generative Description (henceforth FGD) as its theoretical background [7]. In FGD, valency is related primarily to the tectogrammatical layer, i.e., the layer of linguistically structured meaning. The valency characteristics are encoded in a form of a valency frame, which is modeled as a sequence of frame slots corresponding to valency complementations of a verb (labeled by (rather coarse-grained) tectogrammatical roles as 'Actor', 'Patient', 'Effect', 'Direction', etc. [5]). In addition, possible morphemic forms are specified for each valency complementation. For our purposes, we enhance FGD (i) with the concept of lexical-conceptual structures [6], representing lexical-semantic properties of verbs, and (ii) with the open set of labels for situational participants (as 'Agent', 'Recipient', 'Filler', 'Surface', etc.).

The paper is structured as follows: In Section 2, we define the notions situation and perspective, which play a crucial role in the characteristics of diatheses. Then on the basis of the correspondence between situational participants, valency complementations and surface syntactic positions, we distinguish two types of diatheses: grammatical diatheses (Section 3) and semantic diatheses (Section 4). In Section 3.1 and 4.1, the representation of grammatical and semantic diatheses in the valency lexicon is proposed, respectively. Conclusion and an outlook for future work is presented in Section 5.

2 Situation vs. Perspective

The members of both types of diatheses are usually characterized as constructions denoting the same situation, though, each time from a different perspective. Thus the concepts situation and perspective play a key role in the characteristics of diatheses.

First, let us focus on the concept of a **situation**. The term does not refer to a real-life situation, it is rather a situation modeled by language, i.e., a linguistic situation. The linguistic situation related to an event represents a set of facts and entities, i.e., participants, linked in a unified structure. Thus an analysis of a particular situation denoted by the verb must involve not only the specification

[1] `http://ufal.mff.cuni.cz/vallex/2.5/`

of the relevant number of its participants but also the description of the relations between them, see e.g. [3]. For example, the situation portrayed by the uses of the verb *to smear* in examples (1) and (2) consists of three participants labeled as 'Agent', 'Cover' and 'Surface', and it may be informally described as 'an Agent covers a Surface of an object with a Cover'. We refer to this part of the verbal meaning as a **situational meaning** and to its components as **situational participants**. Situational meaning represents an abstract model of situation which has not yet been linguistically structured.

Sentences expressing the same situational meaning can be usually structured in several ways (i.e., different situational participants can occupy the syntactically prominent positions of subject and direct object). This results in different **perspectives** from which the situation is viewed, see e.g. [2]. In case of the verb *to smear*, the situation can be viewed from the perspective of 'Agent', (i.e., *Peter*), as in (1a) and (1b), from the perspective of 'Cover' (i.e., *butter*), as in (2a), or from the perspective of 'Surface' (i.e., *bread*), as in (2b). The different perspectives in these sentences are manifested by grammatical and lexical-semantic means.

As a result, we distinguish two typologically different changes in the verbal valency structure. **Grammatical diatheses** refer to the relation between the uses of verbs characterized by the differences in the mapping of valency complementations and surface syntactic positions, as in (1a)-(2a) and (1b)-(2b). These differences are based on grammatical means (Section 3). In contrast, **semantic diatheses** refer to the relation between the uses of verbs characteristic of the different correspondence between situational participants and valency complementations, as in (1a)-(1b) and (2a)-(2b). These differences are expressed by lexical-semantic means, i.e., by the change of lexical unit (Section 4).

3 Grammatical Diatheses

Sentences related by a grammatical diathesis express the same situational meaning; however, they are characterized by different perspective which results from the changes in the mapping between valency complementations and surface syntactic positions. Further, within this type, the linking of situational participants and valency complementations remains unchanged. This can be illustrated by the pairs of examples (1a)-(2a) and (1b)-(2b), the first one repeated here as (3).

(3) a. *Peter smeared butter on bread.* – b. *Butter was smeared on bread (by Peter).*

Grammatical diatheses are connected with the morphological category of verbal voice, see esp. [4]. Based on the unmarked/marked form of the verb (with respect to the category of voice), a member of a grammatical diathesis is considered to be unmarked or marked, respectively. Grammatical diatheses are not restricted to well-delimited semantic classes of verbs. They rather relate to a great number of verbs with similar syntactic properties regardless of their semantic class membership. The members of grammatical diatheses satisfy the following criteria:

1. The use of the marked member of grammatical diatheses is conditioned by the grammatical meaning of the verb (represented by a specific verbal grammateme in FGD [4]). In Czech, verbal forms of these marked members typically consist either of auxiliaries and non-finite form of lexical verb or they have a reflexive form.
2. The marked member of grammatical diatheses is prototypically connected with the shift of some of situational participants from the prominent surface syntactic position of subject to a less prominent syntactic position.
3. The correspondence between situational participants and valency complementations remains unchanged; thus the set of situational participants is directly encoded in the valency frame by a sequence of valency complementations. It implies that the number of valency complementations and their type are preserved and the changes in the valency frame are limited only to the changes in morphemic forms of the valency complementations.

The asymmetry between the mapping of the situational participants and the surface syntactic positions corresponding to the verb *to smear* in example (3) is illustrated in Figure 1.

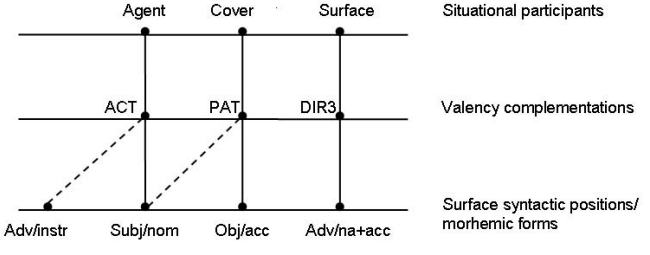


Fig. 1. The changes in the mapping of the valency complementations and the syntactic positions of the verb *to smear* associated with the passive grammatical diathesis

3.1 Representation of Grammatical Diatheses

In this section, we propose the representation of grammatical diatheses in the valency lexicon VALLEX. The changes in valency structure associated with grammatical diatheses are regular enough to be described by general syntactic rules; these rules are stored in the **grammar component** of the lexicon. Then it is sufficient to indicate the applicability of a certain rule in a special attribute attached to each relevant lexical unit of a verb in the **data component** of the lexicon.

Let us demonstrate our approach on the example of the **recipient diathesis**. The marked construction (as in (5)) is characterized by the verbal form consisting of the auxiliary *dostat* 'to get' and the past participle of a lexical verb. The structural condition on the recipient diathesis is the presence of a valency complementation expressed by the dative case in the valency frame corresponding

to the situational participant 'Recipient'. The following verbs satisfy this condition: *doporučit* 'to recommend', *nahradit* 'to recompense', *nařídit* 'to command', *nařezat* 'to thwack', *přidělit* 'to allocate', *vynadat* 'to rebuke', *zaplatit* 'to pay', etc. We propose the following representation of the recipient diathesis in the lexicon:

(i) A formal syntactic rule describing the changes in the valency structure of verbs is stored in the grammar component of the lexicon.
(ii) In the data component, both verb uses are represented by a single lexical unit described by an unmarked valency frame. The applicability of the formal syntactic rule is ascribed to each relevant lexical unit.

This approach is exemplified by the verb *přidělit* 'to allocate', sentences (4)-(5). The valency structure of this verb consists of three valency complementations: obligatory ACT ('Actor') expressed by the nominative case, obligatory ADDR ('Addressee') expressed by the dative, and obligatory PAT ('Patient') in the accusative. ADDR corresponds to the situational participant 'Recipient'. In the marked construction of the recipient diathesis (5), ACT is shifted from the subject position into the less prominent position of an adverbial. The vacated position of the subject is filled by the valency complementation ADDR. The remaining valency complementation PAT stays in the same syntactic position.

(4) *Ministerstvo kultury.*ACT_{nom} *přidělilo obci.*ADDR_{dat} *dotaci.*PAT_{acc} *na opravu kostela.*
Eng. The Ministry of Culture.ACT has awarded a grant.PAT for the repair of the church to the village.ADDR.

(5) a. *Obec.*ADDR_{nom} *dostala přidělenu dotaci.*PAT_{acc} *na opravu kostela (od Ministerstva kultury).*ACT_{od+gen}
Eng. The village.ADDR has been awarded a grant.PAT for the repair of the church (by the Ministry of Culture).ACT

The shifts of the valency complementations ACT and ADDR are manifested by changes in their morphemic forms. This can be described by a syntactic rule Recip.r.

Commentary on the Recip.r
(1) The specific verbal meaning, underlying the use of the marked member of recipient diathesis, is represented by the verbal grammateme 'Recip': its value for the unmarked construction is 0 and for the marked construction, it is 1.

Table 1. Recip.r rule for the recipient diathesis

Recip.r	Unmarked	Marked	Note
verbal grammateme	Recip: 0	Recip: 1	(1)
valency frame	ACT_{nom}	ACT_{od+gen}	(2)
	ADDR_{dat}	ADDR_{nom}	(3)

(2) The shift of the valency complementation ACT from the subject position into the adverbial position is manifested by the change of its morphemic form from the nominative into the prepositional group *od*+genitive.
(3) The shift of the valency complementation ADDR from the indirect object position into the prominent subject position is expressed by the change of its morphemic form from the dative into the nominative.
(4) Every valency complementation that is not listed in the rule is preserved.

For example, if we apply the rule Recip.r to the valency frame describing the unmarked use of the verb *přidělit* 'to allocate', example (4), we derive the valency frame corresponding to the verb in the marked construction of the recipient diathesis, example (5), as follows:

$$\mathrm{ACT}_{nom}\ \mathrm{ADDR}_{dat}\ \mathrm{PAT}_{acc} \Rightarrow_{\text{Recip.r}} \mathrm{ACT}_{od+gen}\ \mathrm{ADDR}_{nom}\ \mathrm{PAT}_{acc}$$

Other grammatical diatheses may be described in the same way:

Passive diathesis
Zaměstnanci informovali vedení podniku o stávce. – Vedení podniku bylo zaměstnanci informováno o stávce.
Eng. The employees have informed the top management about the strike. – The top management has been informed about the strike by the employees.

Deagentive diathesis
Dělníci stavějí novou školu. – Staví se nová škola.
Eng. The workers build a new school. – '(they) build - *Refl* - new - school$_{acc}$' (= A new school has being built.)

Resultative diathesis
Matka uvařila babičce oběd. – Babička má uvařen oběd (od matky).
Eng. Mother has prepared lunch for the grandmother. – 'Grandmother$_{nom}$ - has - prepared - lunch - (by mother)' (= Grandmother has got lunch (prepared by mother).)

Dispositional diathesis
Učím se matematiku. – Matematika se mi učí dobře.
Eng. I learn math. – 'Math$_{nom}$ - *Refl* - me$_{dat}$ - learn - well.' (= Mathematics is easy for me to learn.)

4 Semantic Diatheses

Sentences related by a semantic diathesis express the same situational meaning, similarly as grammatical diatheses. However, in case of semantic diatheses, the different perspective is reflected by the changes in the mapping of situational participants and valency complementations. This can be illustrated by examples (1a)-(1b) and (2a)-(2b), the first one repeated here as (6).

(6) a. *Peter smeared butter on bread.* – b. *Peter smeared bread with butter.*

In contrast to grammatical diatheses, semantic diatheses are not connected with changes of grammatical categories of verbs. They are rather related to a small number of well-delimited semantic classes of verbs which share certain facets of meaning. The members of semantic diatheses satisfy the following criteria:

1. Semantic diatheses are expressed by lexical-semantic means, i.e., by different lexical units. The members of semantic diatheses do not differ from each other in a specific grammatical meaning of a verb; instead, they differ in structuring situational participants into a valency frame.
2. Semantic diatheses are characterized by shifts of some of situational participants from the prominent surface syntactic position of object or subject to a less prominent syntactic position.
3. The changes in a verbal valency structure arisen from the changes in the correspondence between situational participants and valency complementations may affect the number of valency complementations, their type as well as their morphemic form(s).

The asymmetry between the correspondence of the situational participants and the valency complementations of the verb *to smear* in example (6) is illustrated in Figure 2.

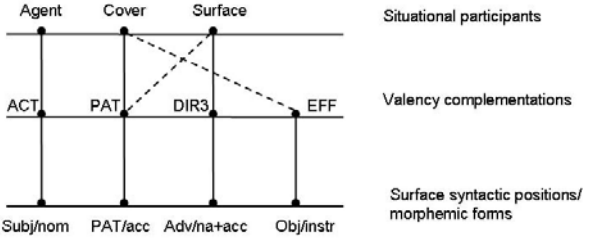


Fig. 2. The changes in the mapping of the situational participants and the valency complementations of the verb *to smear* associated with the semantic diathesis

4.1 Representation of Semantic Diatheses

In this section, we propose the representation of semantic diatheses in the valency lexicon VALLEX. As the members of semantic diatheses differ in the correspondence between situational participants and valency complementations, an appropriate lexical-semantic representation of the situational meaning of the verb is necessary for their adequate description. For this purpose, we adopt the lexical-conceptual structures proposed in [6].

Furthermore, contrary to grammatical diatheses, the changes in the verbal valency structure associated with semantic diatheses vary even within a single type of diathesis. It follows that they cannot be described by general syntactic

rules. For these reasons, we propose to specify separate lexical units corresponding to the individual members of semantic diatheses in the **data component** of the lexicon; these lexical units are interlinked by a relevant type of semantic diathesis. In the **grammar component**, the changes in the verbal valency structure are represented by lexical rules. Our approach can be explained on the example of the **locative semantic diathesis**, see below. Let us mention some other types of Czech semantic diatheses that may be described in the same way:

Material-Product diathesis
*Nařezal kládu.*PAT-Material *na tři polena.*EFF-Product.
Eng. He cut the log.PAT-Material into three pieces.EFF-Product
*Nařezal tři polena.*PAT-Product *z klády.*ORIG-Material
Eng. He cut three pieces.PAT-Product from the log.ORIG-Material
Source-Substance diathesis
*Slunce.*ACT-Source *vyzařuje teplo.*PAT-Substance
Eng. The sun.ACT-Source radiates heat.PAT-Substance
*Teplo.*ACT-Substance *vyzařuje ze Slunce.*DIR-Source
Eng. Heat.ACT-Substance radiates from the sun.DIR-Source
Agent-Location diathesis
*Včely.*ACT-Agent *se hemží na zahradě.*LOC-Location
Eng. Bees.ACT-Agent are swarming in the garden.LOC-Location
*Zahrada.*ACT-Location *se hemží včelami.*PAT-Agent
Eng. The garden.ACT-Location is swarming with bees.PAT-Agent

Representation of Locative Semantic Diathesis. Whereas grammatical diatheses are primarily conditioned by syntactic properties of verbs, semantic diatheses are rather associated with semantic characteristics of verbs. For example, the locative diathesis is typical of the verbs denoting 'co-occurrence' in a broad sense. This class of verbs is, however, semantically heterogeneous: some verbs indicate 'creating co-occurrence', e.g., *naložit* 'to load', *natočit* 'to draw', *namazat* 'to smear', whereas others express 'destroying co-occurrence', e.g., *setřít* 'to wipe', *sklidit* 'to clear', *vybrat* 'to pick out'. Moreover, the verbs from both these subclasses may be distinguished according to whether they express the relation 'inside' (connected with the situational participants 'Container' and 'Filler') or 'outwards' (the situational participants 'Surface' and 'Cover'). Then all these verbs are characterized by the ability to linguistically structure the meaning 'co-occurrence' in two different ways, see (7) and (8).

(7) *Farmáři.*ACT-Agent *naložili seno.*PAT-Filler *na vůz.*DIR-Container
Eng. The farmers.ACT-Agent loaded hay.PAT-Filler on the truck.DIR-Container

(8) *Farmáři.*ACT-Agent *naložili vůz.*PAT-Container *senem.*EFF-Filler
Eng. The farmers.ACT-Agent loaded the truck.PAT-Container with hay.EFF-Filler

We propose the following representation of the locative diathesis in the valency lexicon:

(i) In the data component, the members of semantic diathesis are represented by separate lexical units, which are interlinked by a relevant type of semantic diathesis.
(ii) In the grammar component, the changes in valency structure of verbs are captured by a lexical rule determining the changes in the mapping of situational participants and valency complementations.

Locative Diathesis of Verbs Indicating 'Creating Co-occurrence'. Let us demonstrate our approach on the example of the verb *naložit* 'to load'. With respect to its semantic properties, we determine the following three situational participants: 'Agent', 'Filler' and 'Container'. The situational meaning of the verb may be informally described as 'an Agent fills a Container with a Filler'. This meaning is syntactically structured in two ways, as in (7) and (8) above.

Both sentences (7) and (8) express the change of location of the situational participant 'Filler' caused by the participant 'Agent'. In comparison with the variant (7), variant (8) is semantically more complex – it implies, in addition, the change of state of the participant 'Container'. This change of state is associated with a holistic interpretation and it results from the change of location of the 'Filler'; i.e., variant (8) implies that the 'Container' is full of the 'Filler', in contrast to variant (7), see esp. [1].

Now let us formulate the lexical-conceptual structures (henceforth LCSs) representing the uses of the verb *naložit* 'to load' related by the locative semantic diathesis. The LCSs must necessarily reflect the situational meaning common to both uses as well as their semantic differences. The LCSs in (a) and (b) correspond to examples (7) and (8), respectively:

(a) [[x ACT$<_{LOAD}>$] CAUSE [BECOME [y INTO z]]]
(b) [x CAUSE [BECOME [z $<$*LOADED*$>$]] BY MEANS OF [[x ACT$<_{LOAD}>$] CAUSE [BECOME [y INTO z]]]]

Commentary on the LCSs. (a) This LCS represents a complex event – change of location – consisting of two subevents. (i) The first subevent represented as [x ACT$<_{LOAD}>$] identifies the action of the 'Agent'. The verb $<$*LOAD*$>$ in the subscript serves as a modifier of the action. This modifier specifies a way the action is carried out. (ii) The second part of the LCS [BECOME [y INTO z]] represents the change of location of the 'Filler' resulted from the first subevent, see the predicate CAUSE. The preposition INTO identifies the semantic modification of the locative diathesis 'creating co-occurrence' with the relation 'inside'. The labels of the situational participants are associated with the position of the variables in the LCS as follows: x $\sim$ 'Agent', y $\sim$ 'Filler', and z $\sim$ 'Container'. (b) In comparison with the LCS (a), the LCS (b) is more complex. In addition to LCS (a), it contains the component [BECOME [z $<$*LOADED*$>$]] specifying the change of state of the 'Container' indicated as $<$*LOADED*$>$. Relating the component [BECOME [z $<$*LOADED*$>$]] with the whole LCS (a) indicates that this event arises as a consequence of the event identified by the LCS (a). In

the LCS (b), the same correspondence between the variables and the labels is preserved as in LCS (a).

With respect to the complexity, we consider the variant corresponding to the LCS (a) as unmarked and the variant characterized by the LCS (b) as the marked one.

Table 2. The possible mapping of the situational participants and the valency complementations of the verbs expressing 'creating co-occurrence' specifying the relation 'inside'

x ∼ 'Agent'	**y ∼ 'Filler'**	**z ∼ 'Container'**	examples
ACT	PAT	DIR	*naložit seno na vůz* *nasypat mouku do pytle*
ACT	EFF	PAT	*naložit vůz senem*
ACT	∅	PAT	*nasypat pytel *moukou*

Considering the mapping between the valency complementations and the situational participants represented by the variables in the LCS (a) and LCS (b) (Table 2), we formulate the lexical rule Loc.r1 for the locative diathesis. The rule Loc.r1 can be applied also to other verbs expressing 'creating co-occurrence' specifying the relation 'inside', e.g., *natočit* 'to draw', *nasypat* 'to pour', *doplnit* 'to add' (as well as to those verbs specifying the relation 'outwards', see below):

'inside'	'outwards'	LCS(a)/LCS(c)		LCS(b)/LCS(d)
Filler	Cover	PAT	$\Rightarrow_{\text{Loc.r1}}$	EFF / ∅
Container	Surface	DIR	$\Rightarrow_{\text{Loc.r1}}$	PAT

Commentary on the Loc.r1. On the left side of the rule, the valency complementations of the unmarked member of the diathesis are given, i.e., the situational participant mapped onto PAT in the valency frame of the verb represented by the LCS (a) (and LCS (c) below) is changed into EFF in the frame corresponding to the LCS (b) (and LCS (d)). If EFF is not present in the valency frame, then this participant is not linguistically structured, see ∅ in Table 2. The situational participant mapped onto the valency complementation DIR in the valency frame of the unmarked member of the diathesis corresponds to the valency complementation PAT in the frame of the marked member.

The rule Loc.r1 holds also for the changes in the mapping of situational participants and valency complementations of the verbs indicating 'creating co-occurrence' with the relation 'outwards', e.g., *natřít* 'to smear' and *namalovat* 'to paint'. Situational meaning of these verbs is represented similarly as for the verb *naložit* 'to load'. See the LCSs (c) and (d) representing the verb *natřít* 'to smear' in examples (9) and (10), respectively:

(9) *Petr*.ACT-Agent *natřel barvu*.PAT-Cover *na zeď*.DIR-Surface
Eng. Peter.ACT-Agent smeared the paint.PAT-Cover on the wall.DIR-Surface.

(10) *Petr.*ACT-Agent *natřel zeď.*PAT-Surface *barvou.*EFF-Cover
Eng. Peter.ACT smeared the wall.PAT-Surface with paint.EFF-Cover

(c) [[x ACT<$_{SMEAR}$>] CAUSE [BECOME [y ON z]]]

(d) [x CAUSE [BECOME [z <*SMEARED*>]] BY MEANS OF [[x ACT<$_{SMEAR}$>] CAUSE [BECOME [y ON z]]]]

However, in case of the verbs expressing 'creating co-occurrence' with the relation 'outwards', another set of labels for the situational participants is associated with the variables in the LCS (c) and LCS (d): x ∼'Agent', y ∼'Cover', and z ∼ 'Surface'. Despite the different set of labels, the changes in the mapping of the situational participants and the valency complementations are described by the same rule Loc.r1 as the changes of the verbs indicating the relation 'inside'.

Locative Diathesis of Verbs Indicating 'Destroying Co-occurrence'. For the description of the changes in the correspondence between situational participants and valency complementations of the verbs expressing the event 'destroying co-occurrence', we formulate the lexical rule Loc.r2. We demonstrate this rule on the example of the verb *očistit* 'to clean', see (11) and (12).

(11) *Jana.*ACT-Agent *očistila bláto.*PAT-Cover *z bot.*DIR-Surface
Eng. Jane.ACT-Agent cleaned mud.PAT-Cover of the shoes.DIR-Surface

(12) *Jana.*ACT-Agent *očistila boty.*PAT-Surface *od bláta.*ORIG-Cover
Eng. Jane.ACT cleaned the shoes.PAT-Surface of mud.ORIG-Cover

With respect to the complexity of the events, we consider the use in (11) as the unmarked one, see the corresponding LCS (e), whereas the use in (12), represented by the LCS (f), as the marked one. The labels of the situational participants are identified with the positions in the LCSs as follows: x ∼ 'Agent', y ∼ 'Cover', and z ∼ 'Surface':

(e) [[x ACT<$_{CLEAN}$>] CAUSE [BECOME [y OF z]]]

(f) [x CAUSE [BECOME [z <*CLEANED*>]] BY MEANS OF [[x ACT<$_{CLEAN}$>] CAUSE [BECOME [y OF z]]]]

The lexical rule Loc.r2 describes the changes in the correspondence of the situational participants and the valency complementations of the verb *očistit* 'to clean'. This rule can be applied also to other verbs indicating 'destroying co-occurrence' expressing both relations 'outwards', as e.g., *sklidit* 'to clear', *setřít* 'to wipe', *oloupat* 'to peel off', and 'inside', e.g., *vybrat* 'to pick out', *vyklidit* 'to clean out' (in the latter case, the LCS variables are associated with the labels for the situational participants as follows: x ∼ 'Agent', y ∼ 'Filler', and z ∼ 'Container').

'outwards'	'inside'	LCS(e)		LCS(f)
Cover	Filler	PAT	$\Rightarrow_{\text{Loc.r2}}$	ORIG / ∅
Surface	Container	DIR	$\Rightarrow_{\text{Loc.r2}}$	PAT

Commentary on the Loc.r2. On the left side of the rule, the set of valency complementations representing the unmarked member of the diathesis is given. The situational participant 'Cover' or 'Filler' mapped onto PAT in the valency frame represented by the LCS (e) is mapped onto ORIG in the frame represented by the LCS (f). If ORIG is not present in the marked frame, then this participant is not expressed. The situational participant 'Surface' or 'Container' corresponding to the valency complementation DIR in the valency frame of the unmarked member of the diathesis, described by the LCS (e), is mapped onto the valency complementation PAT in the frame of the marked use, represented by the LCS (f).

5 Conclusion and Future Work

We have distinguished two types of relations between semantically close uses of verbs, which are syntactically structured in different ways: grammatical and semantic diatheses. We have proposed their representation in the valency lexicon VALLEX. The changes in a verbal valency structure associated with grammatical diatheses are described by formal syntactic rules which determine regular changes in morphemic form(s) of complementations. Thus both verbal uses may be represented by a single lexical unit, with ascribed information on applicability of individual formal syntactic rule for a relevant grammatical diathesis.

In contrast, the changes typical of semantic diatheses are represented by lexical rules which formally describe the changes in the mapping of situational participants and valency complementations. It implies that both verbal uses are represented by separate lexical units interlinked by a relevant type of semantic diathesis. In the future, we intend to represent typologically different changes in valency structure of verbs in the lexicon in a similar way.

References

1. Anderson, S.R.: On the Role of Deep Structure in Semantic Interpretation. Foundations of Language (7), 387–396 (1971)
2. Fillmore, C.J.: The Case for Case Reopened. In: Form and Meaning in Language, pp. 175–199. CSLI Publications, Stanford (2003)
3. Mel'čuk, I.A.: Actants in Syntax and Semantics. Language 12, 1–66 (2004)
4. Panevová, J., et al.: Syntax současné češtiny (na základě anotovaného korpusu). Nakladatelství Karolinum, Praha (manuscript)
5. Panevová, J.: Valency Frames and the Meaning of the Sentence. In: Luelsdorff, P.A. (ed.) The Prague School of Structural and Functional Linguistics, pp. 223–243. John Benjamins Publishing Company, Amsterdam (1994)
6. Rappaport Hovav, M., Levin, B.C.: Building Verb Meanings. In: Butt, M., Geuder, W. (eds.) The Projection of Arguments. Lexical and Compositional Factors, pp. 97–134. SLI Publications, Stanford (1998)
7. Sgall, P., Hajičová, E., Panevová, J.: The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Reidel, Dordrecht (1986)

Symbolic Classification Methods for Patient Discharge Summaries Encoding into ICD

Laurent Kevers and Julia Medori

CENTAL - Université catholique de Louvain (UCL)
1, Place Blaise Pascal - 1348 Louvain-la-Neuve - Belgium
{laurent.kevers,medori.julia}@uclouvain.be

Abstract. This paper addresses the issue of semi-automatic patient discharge summaries encoding into medical classifications such as ICD-9-CM. The methods detailed in this paper focus on symbolic approaches which allow the processing of unannotated corpora without any machine learning. The first method is based on the morphological analysis (MA) of medical terms extracted with hand-crafted linguistic resources. The second one (ELP) relies on the automatic extraction of variants of ICD-9-CM code labels. Each method was evaluated on a set of 19,692 discharge summaries in French from a General Internal Medicine unit. Depending on the number of suggested classes, the MA method resulted in a maximal F-measure of 28.00 and a highest recall of 46.13%. The best F-measure for the second method was 29.43 while the maximal recall was 52.74%. Both methods were then combined. The best recall increased to 60.21% and the maximal F-measure reached 31.64.

Keywords: Patient discharge summaries, ICD-9-CM, classification, symbolic approach.

1 Introduction

This paper presents work done in collaboration with one of the major hospitals in Brussels: *les Cliniques universitaires Saint-Luc*. The Belgian government requires hospitals to report their activity through the encoding of patient's stays into ICD-9-CM[1]. Codes from this classification symbolise diagnoses, procedures, aggravating factors such as allergies, smoking, but also elements in the patient's past history that may influence his/her current health status.

The encoding task is generally done by professional coders. Their work consists in going through the patient's medical record and 'translating' into ICD-9-CM codes every activity that occurred during the patient's stay. The main source of information coders use is patient discharge summaries (PDS) that physicians write after each patient's stay. These documents are written in free text, in the form of a letter addressed to the patient's GP or external care professionals. The encoding task is a tedious and demanding task that requires very specialized

[1] International Classification of Diseases - Ninth revision - Clinical Modification.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 197–208, 2010.

skills. Consequently, many hospitals try to reduce the amount of work involved by trying to (partly) automate this process.

Our goal is to build a tool that would help coders by providing them with a set of most likely codes. Indeed, the full automation of the task is difficult to achieve as the PDS seldom contains all the necessary information[2].

The automation of the encoding process can be considered as analogous to a classification problem. It involves classifying PDS into a nomenclature. Here, ICD-9-CM codes are considered as classes. There are two main approaches to this classification problem: symbolic and statistical methods. Statistical approaches are based on machine learning methods and require a large amount of annotated data for training, which makes it difficult for this type of approach to face a change of nomenclature (*Saint-Luc* will adopt ICD-10 in the near future) and to classify certain very specific codes for which we have only sparse data available.

In this paper, the aim is therefore to develop a symbolic approach to the encoding task. Symbolic approaches involve linguistic knowledge, are therefore usually more language-dependent and require more time for development. However, in one of our methods (method 2), we will see that it is possible to partly generate the linguistic resources automatically. Two methods are described: method 1 is based on the morphological analysis of medical language; method 2 relies on the automatic extraction of variants of ICD-9-CM code labels. Method 2 is thus limited to the vocabulary used in the resource whereas method 1 uses a wider range of vocabulary. They should therefore complement each other.

After a brief description of related work (section 2) and evaluation data (section 3), sections 4 and 5 will detail the two methods and their respective results. Then, their combination will be discussed in section 6 before considering ways to improve the performance of the system in section 7.

2 Related Work

Since the early 1990s, many scientists have looked into the possible automation of the encoding process [1] [2] [3]. As mentioned above, there are two main approaches to the encoding task: knowledge-based (e.g. MedLEE [4]) and machine learning (e.g. Autocoder [5]) Both approaches scored highly in the 'Computational Medicine Challenge' in 2007 [6]: among the best three systems, two combined a statistic and a symbolic approach [7] and only one relied only on a symbolic approach [8]. All these studies were developed on English language. Pereira et al. [9] built a fully symbolic system for French relying on a linguistic-based indexing system into the French version of the MeSH classification and then mapping it to ICD-10. What is noteworthy is that most systems, even when choosing a statistical approach, still rely on a linguistic component. The results of most of these studies are promising - Autocoder achieved very high precision for two thirds of the assigned codes - but the documents are often more structured

[2] Additional information can be found in the full medical record but to avoid having to deal with the variety of formats in the record, we decided to focus our work on PDS as sources of information.

than our PDS (e.g. diagnoses clearly marked). And they are also often limited with regard to the number of codes involved or to the types of documents.

3 Data

The ICD-9-CM is a hierarchical nomenclature comprising 15,688 **codes**, which are 4 or 5 characters long. The first three characters represent a general **category** of diagnoses, and the next one or two digits specify the exact diagnosis (Fig. 1). The ICD-9-CM is divided into 1,135 general categories. In the perspective of a coding help, our study will classify according to these categories, letting the coder choose the right code into the hierarchy within each suggested category.

Code	Label
001	**Cholera**
0010	Cholera due to Vibrio cholerae
0011	Cholera due to Vibrio cholerae el tor
0019	Cholera, unspecified

Fig. 1. Hierarchical structure of ICD9-CM

Our evaluation data consists in 19,692 PDS in French taken from the General Internal Medicine unit. Patients in this unit suffer from very diverse diseases. A wider range of codes is therefore used (6,029 different codes dispatched into 895 categories). The PDS in our corpus were assigned 150,116 codes (137,336 categories) which makes an average of 7.6 codes (7 categories) per document. Note that 27% (241 out of 895) of the categories were used less than 6 times. These manually assigned codes are used as a *gold standard* for our evaluation.

In order to broaden the scope of our linguistic resources, we used the UMLS[3] as a source of variants for the ICD-9-CM code labels. The UMLS metathesaurus unifies and integrates into one unique resource many medical nomenclatures from different languages, giving to each concept a unique concept identifier (CUI). Using this CUI, we were able to extract from this metathesaurus different wordings for the ICD-9-CM code labels. These 'synonyms' were then added to the original code labels, as illustrated in Fig. 2.

Class	French label	English label	Source
061	Dengue	Dengue	ICD-9-CM
	Dengues	Dengues	UMLS
	Fièvre dengue	Dengue fever	UMLS
	Infection par le virus de la dengue	Infection by the dengue virus	UMLS

Fig. 2. Definition of class '061' with terms from ICD-9-CM and UMLS

[3] Unified Medical Language System, http://www.nlm.nih.gov/research/umls/

4 Classification Method 1: Morphological Analysis (MA)

This method follows a two-part structure to re-create the work of human coders, who first read the PDS while highlighting information that needs to be encoded and then assign codes to these expressions. Therefore, the first module (**extraction module**, section 4.1), already presented in [10], aims at restricting the information to be processed by selecting informative sequences of text. The second module (**encoding module**, section 4.2), is a new one which 'translates' the extracted phrases into codes.

4.1 Extraction Module

This module selects in PDS sequences of text that convey information that needs to be encoded: diagnoses, procedures, allergies, etc. This module is based on specialized dictionaries and transducers built with Unitex[4] [11].

The specialized dictionaries were built partly automatically and completed manually. The main dictionaries are the dictionaries of diagnoses and procedures. These two dictionaries were automatically constructed using the French nomenclatures comprised in the UMLS as described in section 3. As cataloguing all the different ways a physician may mention a diagnosis is difficult, these dictionaries still needed to be completed manually. All the dictionaries were then automatically inflected.

To compensate the lack of exhaustivity of the dictionaries, we needed to use more flexible linguistic resources to detect diagnoses and informative data. We therefore used finite state transducers[5]. Transducers are represented as graphs (see Fig. 3). Each path of the graph represents a recognized sequence. These hand-crafted transducers aim at marking up the phrases that mention a diagnosis or a procedure by adding XML tags to the original text. In Fig. 3, the graph recognizes different ways of mentioning fractures ('fracture') and sprains ('foulures') as well as the body part affected (with a link to the anatomical dictionary in subgraph *d_localisation*). It then outputs the XML tags <MALINDET>, indicating that the phrase is a diagnosis.

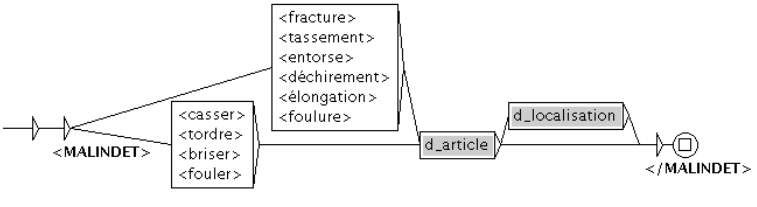


Fig. 3. Transducer for fractures

[4] A corpus processing tool: http://www-igm.univ-mlv.fr/~unitex/

[5] A transducer is not only able to recognize a sequence of elements but also to produce a related output.

When diseases or procedures are not in the dictionaries, graphs allow us to locate these terms thanks to clues in the text: the context in which the term appears and its morphology. Contexts like 'The patient presents with', 'shows signs of ...' indicate that what follows may be a disease or a symptom and will therefore be extracted. The morphological clues consist mainly of word endings like '-pathy', indicating a disease or '-therapy' for a procedure.

Detecting contexts is also very important as it influences the way the pathology will be encoded: a negated diagnosis will not be encoded, or past history illnesses are coded differently.

This extraction module was evaluated in [10] on a corpus of 220 PDS: 66.6% of phrases highlighted manually by the professional coders were also extracted by the system, and 85.5% of the extracted phrases were diagnoses. More work has been done on the system since then but it has not yet been re-evaluated.

4.2 Encoding Module

Once the terms containing information to be encoded are extracted, they need to be matched to an ICD-9-CM code. This classification method aims at taking advantage of the fact that medical language has a rich morphology. When comparing the original terms used in the PDS and the corresponding code label, we observed that many terms looked morphologically close. We therefore developed a methodology very similar to a 'bag-of-words' approach, where the extracted terms and the ICD code labels were matched according to the morphemes composing them. The breaking down into morphemes was done using DériF[6] [13]. In Fig. 4, 'fibroscopie bronchique' and 'bronchoscopie par fibre optique' are both equivalents in French of 'bronchoscopy'. A word-based similarity measure would not have matched these two phrases since they do not share a word in common. But when split into morphemes, these two terms share two components.

However, we are still missing an element. Indeed, 'fibr-' and 'fibre' convey the same meaning 'fiber' but they cannot be matched only by looking at their morphemes. The matching process was therefore extended to the meaning conveyed by the morphemes. Meanings are available in DériF derivation process. Here, 'fibre' is added to the components of 'fibroscopie bronchique', which adds one more common element between the two terms. Stopwords[7] are removed from the list. The word 'antécédent' (*past history*) is added to phrases extracted with a past history context so as to match preferably the specific past history codes. Terms extracted with a negative context were not taken into account.

The number of common components is then normalised by dividing it by the overall number of elements, so as not to favour long strings. This gives us a similarity value ($S(T_iC_j)$) between one extracted term (T_i) and one code

[6] Dérivation en Français, a morphosemantic analyser adapted to medical language in [12].

[7] In this experiment, stopwords are words as well as morphemes that do not convey any meaning (like '-ique' here, equivalent to the English '-ic'), or non-informative phrases such as 'sans autre précision' ('unspecified').

Fibroscopie bronchique (PDS)	Bronchoscopie par fibre optique (ICD-9-CM)
fibr-	**bronch-**
-scopie	**-scopie**
bronch-	fibre
-ique	optique

Fig. 4. Example of bronchoscopy

label (C_j): $S(T_iC_j) = \frac{N_{T_i \cap C_j}}{(N_{T_i}+N_{C_j})}$ where $N_{T_i \cap C_j}$ is the number of common words between T_i and C_j, N_{T_i} and N_{C_j} are the number of words on each side.

The resulting bag-of-word of each extracted phrase is then compared to all the code labels and their 'synonyms'. To assign codes to the entire PDS, the maximum similarity value for each code is kept: $Score(C_j) = max(S(T_iC_j))$.

All the codes are ordered according to their similarity value. The first codes in this list are then considered as the most likely codes to be assigned to the document. The score for each parent category is the maximum score of its children codes. The output list is then ranked according to this parent category score. This list can be returned as it is or shortened using a thresholding function [14].

4.3 Results

The encoding evaluation was conducted on 19,692 documents. The measures, Recall, Precision and F-measure (F_1), were macro-averaged[8]. For a document, R gives the proportion of manually assigned classes retrieved in the suggested list and P the proportion of good classes, as defined by the *gold standard*, into this list. The best results are reported on table 1. Depending on the will to promote recall or, on the contrary, precision, we can tune the thresholding function acceptance level and then compute intermediate results. Of course a higher recall level always comes with a higher number of suggested categories.

Table 1. Evaluation of MA method

	Recall (R)	Precision (P)	F-measure (F_1)	Nb. classes	Threshold
Best Recall	**46.13**	14.70	21.10	20	No
Best F-measure	34.52	27.34	**28.00**	8.6	Yes

5 Method 2: Extended Lexical Patterns (ELP)

This method, previously described in [14] and used in [15] on parliament documents, is based on the insight that well described classes[9] can be sufficient to find a significant intersection with the text vocabulary. Class labels are extracted from

[8] Computed for each document and then averaged.

[9] Class defined by a descriptive set of words and/or compound expressions.

existing terminological resources (e.g. thesauri, nomenclatures). For this paper, the ICD-9-CM enhanced with the UMLS 'synonyms' was used (section 3).

Our approach attaches value to compound expressions because of their high descriptive power. They are often used to refer to complex concepts or objects and as a result are good items to contribute to class definition.

The extended lexical patterns (ELP) method consists in two steps. First, the automatic transformation of a class definition resource into a term **extraction resource** (Section 5.1). It detects in each text a list of expressions considered as interesting for class inference. Then the **class assignment** (Section 5.2) step uses this result for classification. The first step is performed once while the second must be repeated for each new document.

5.1 Extraction Resource Building

The original nomenclature is automatically converted to finite state transducers, compatible with Unitex. They are made of: (a) lexical elements from the original class label; (b) other more generic items, like grammatical codes[10] or meta labels[11]. The aim is to increase the coverage with generic elements while preserving the good precision induced by the lexical units. The transducer output is the class related to the recognized sequence.

First, for each category, all IDC-9-CM labels and their UMLS 'synonyms' are gathered. The complex expressions are then automatically cleaned and split. Some recurrent and non-informative parts, like 'sans autre précision' ('unspecified'), are first removed. Splitting operations include acronyms extraction, parenthesis handling and enumerations parsing as illustrated in Fig. 5.

Entérite à petit Virus rond (PVR)	
Enteritis due to other small round viruses (SRV's)	
↪Entérite à petit Virus rond	↪PVR
Hyperparathyroïdie secondaire (d'origine rénale)	
Secondary hyperparathyroidism (of renal origin)	
↪Hyperparathyroïdie secondaire	↪Hyperparathyroïdie secondaire d'origine rénale
Otite moyenne, chronique, suppurative	
Chronic suppurative otitis media	
↪Otite moyenne	↪Otite moyenne, chronique
↪Otite moyenne, chronique, suppurative	
Maladies infectieuses et parasitaires	
Infectious and parasitic diseases	
↪Maladies infectieuses	↪Maladies parasitaires
↪Maladies infectieuses et parasitaires	

Fig. 5. Examples of splitting operations

[10] Example: <N> for nouns, <A> for adjectives, <V> for verbs...

[11] Example: <TOKEN> recognizes any token.

The second step is stopwords processing. They are not removed but replaced with meta labels, like <TOKEN>[12]. This substitution improves the coverage to similar expressions where small vocabulary variations occur.

ORIGINAL FORM: **Atteinte d'un nerf** - Injury to nerves
MODIFIED PATTERN: Atteinte <TOKEN> nerf
EXAMPLE OF RECOGNIZED FORMS: Atteinte (du|de|d'un|des) nerf

Stemming is also used to broaden the recognition. Snowball[13] [16] provided stems used to build regular expressions[14]. The patterns are then able to recognize expressions with words beginning with these stems. Grammatical variations are therefore covered: gender/number agreement, noun form/adjectival form swap, etc. Note that we also remove accents, and decapitalize letters.

ORIGINAL FORM: **Oedème de membre inférieur** - Edema of legs
MODIFIED PATTERN: <<^oedem>> <TOKEN> <<^membr>><<^inferieur>>
EXAMPLE OF RECOGNIZED FORM: oedeme**s** de**s** membre**s** inferieur**s**

The next step aims at allowing more important variations: additional words may appear (e.g. adjectives) but also other signs (comas, parentheses, etc.). To do this, we allow any token to be inserted between two elements of the transducer. This is achieved with a special 'insert' subgraph which contains a <TOKEN> tag. Figure 6 shows the final transducer automatically generated for class '061'.

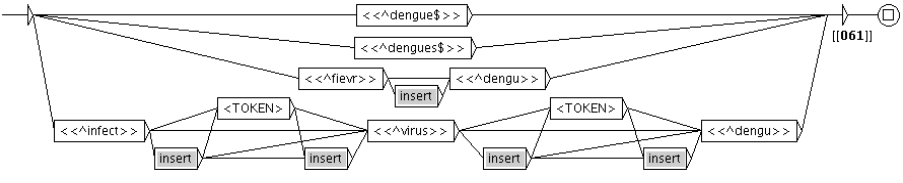


Fig. 6. Final transducer for class '061'

All transducers are gathered into a main transducer which also contains some additional elements to locate negative contexts and to avoid the assignment of classes to expressions such as 'absence d'infection' ('no infection').

5.2 Class Inference

Once the extraction resource is available, it is applied to each new document. The result of this operation is a list of expressions[15] (see Fig. 7). Each item of the list comes with a class identifier ([[***]]), and in the case of a negative context the minus tag ([[-]]) is added.

[12] Note that a <TOKEN> tag is not allowed on its own or at the begin or at the end of a pattern.
[13] An implementation of the Porter algorithm: http://snowball.tartarus.org/
[14] The signs '<<' and '>>' determine the regular expression and ^specify the anchor at the beginning of the string.
[15] An expression can of course occur more than once and can also be linked to more than one class. In this last case, the expression participate to the score of each class.

zona [[053]]	oesophagite moderee aspecifique [[947]]
extremement douloureux [[729]]	infection a mycobacterie [[031]]
gastroscopie [[Z44]]	fond de oeil [[Z16]]
acide [[E96]]	pas de [[-]] atteinte du nerf [[957]]
anemie normochrome normocytaire [[285]]	zona [[053]]
sequellaires apicales droite (tuberculose [[137]]	hyperthyroidie [[242]]
intestin grele [[Z45]]	goitre [[706]]
tuberculose [[V12]]	goitre [[240]]

Fig. 7. List of recognized expressions from one text

For each expression, a weight based on its frequency is computed[16]. The frequency measure is multiplied by 2 in the case of a multi-word expression. A final weight for each represented category is then obtained after the addition of all related expression weights. This list can be returned as it is or shortened by a thresholding function [14].

5.3 Results

The evaluation was conducted similarly as for the first method (section 4.3). The best results are reported on table 2. As in section 4.3, intermediate results can be computed depending on the interest to promote recall or precision.

Table 2. Evaluation of ELP method

	Recall (R)	Precision (P)	F-measure (F_1)	Nb. classes	Threshold
Best Recall	**52.74**	20.69	27.37	19.6	No
Best F-measure	37.97	30.30	**29.43**	9.8	Yes

6 Combination of Symbolic Methods

The two methods detailed here can be used on their own. However their combination may improve the classification results. In order to test that hypothesis, we implemented the same mixing methodology as in [15]. This process consists in the merging of the resulting lists of classes by computing their weighted[17] union (Mix1) or intersection (Mix2) before a possible thresholding. To be comparable, the weights were normalized (values between 0 and 1). The list of classes can also be thresholded before their union (Mix3) or intersection (Mix4). These last two approaches are based on the lists of classes returned by each method to maximize F_1 (see sections 4.3 and 5.3)[18].

[16] The use of a *TF-IDF* did not bring significant improvement as in previous work [14].

[17] An hyperparameter α balances the importance of each method: $merged = \alpha * method1 + (1 - \alpha) * method2$ with $0 \leq \alpha \leq 1$ and steps of 0.1.

[18] In these cases, the use of the hyperparameter has no influence on the results because thresholding comes before balancing.

6.1 Results

Combined methods results are presented in table 3[19]. This approach clearly improves both recall and f-measure. Generally, the union combination tends to improve the recall while the intersection combination or the thresholding process tend to increase precision. In our experiments, the best results were always reached with the union of unshortened lists (Mix1). For F_1 maximization, thresholding has to be done afterwards.

Table 3. Evaluation of symbolic methods combination

	Recall (R)	Precision (P)	F-measure (F_1)	Nb. classes	Threshold	α / 1-α
Mix1: Threshold(Method$_1$ $\cup$ Method$_2$)						
Best R	**60.21**	13.20	20.86	30.5	No	Any
Best F_1	37.13	33.12	**31.64**	8.1	Yes	0.3 / 0.7
Mix2: Threshold(Method$_1$ $\cap$ Method$_2$)						
Best R	38.66	29.28	30.52	9.1	No	Any
Best F_1	34.73	34.55	31.50	7	Yes	0.3 / 0.7
Mix3: Threshold(Method$_1$) $\cup$ Threshold(Method$_2$)						
Best F_1	43.28	20.59	27.90	14.7	Yes	N/A
Mix4: Threshold(Method$_1$) $\cap$ Threshold(Method$_2$)						
Best F_1	24.07	37.95	29.46	4.4	Yes	N/A

The best recall increases to 60.21% (Mix1) compared with 46.13% (+14.08) for method 1 and 52.74% (+7.47) for method 2. As for method 1 and method 2, the best recall is reached when returning all classes from the merged lists, that is 30.5 classes on average (10 additional categories). Among the codes retrieved by the combination method, 64.21% were returned by both methods and the remaining 35.79% by one method or the other. From this result we can conclude that the two methods complement each other well.

At 31.64 (Mix1), the best f-measure improvement is not as clear-cut but it nevertheless outperforms method 1 (28.00, +3.64) and method 2 (29.43, +2.21). This result is mainly due to the increase of precision for both methods (+5.78% for method 1 and +2.82% for method 2) while the recall remains basically the same for method 2 (-0.84) and improves for method 1 (+2.61). This greater precision reduces the number of suggested categories to 8.1 (previously 8.6 for method 1 and 9.8 for method 2) without lowering the recall level.

The Mix2 approach, which only keeps the intersection of both methods result lists, has lower performance. The highest recall is limited by the proportion of common codes returned by both methods (64.21%) and is therefore lower than for original methods. The maximization of the f-measure gives a similar result (31.50) as for Mix1 and an increase with regard to method 1 and 2. Again, the improvement comes mostly from the precision (+7.2% for method 1 and +4.24% for method 2). However this increase is greater than in Mix1, due to the filter effect of the intersection, and the number of suggested categories dropped to 7.

[19] As a consequence of their implementation (thresholding before merging), we only look at f-measure maximization for Mix3 and Mix4 approaches.

For the last two combinations, regarding to the original methods, Mix3 turns out to be less efficient and Mix4 shows only very little improvements.

Finally, a brief look at the rare codes, *i.e.* those that are used less than 6 times in our test corpus, shows that 35% of their occurrences (212 out of 603) were covered by the (unshortened) list resulting of the union of both methods.

7 Discussion and Future Work

The reported results have to be put into perspective. First, the evaluation was conducted on manual indexing. As shown by several studies, in particular [17] for medical indexing, the maximum inter-annotator agreement is often situated at approximately 70%. Therefore the evaluation of an automatic classification method compared with manual annotation cannot reach the maximal recall and precision. Secondly, we saw that most of the information that needs to be encoded is present in the PDS. However, an internal study in i*Saint-Luc* showed that 15 to 20% of the codes assigned by the coders cannot be inferred from the PDS.

The MA method relies on the extraction of phrases indicating diagnoses and procedures. As indicated in section 4, 66% of the phrases highlighted by professional coders were extracted by the system. This means that 34% of the phrases were not found and therefore no code could be inferred from them. This lowered the maximum recall value accordingly. More time should be dedicated to the development of the graphs used in this module.

Regarding the ELP method, the automatic transformation of the basis resource into the extraction transducer can be improved further. The phrases used to build the transducers have to be as short as possible (to promote recall) and as unambiguous as possible (to promote precision), while some category labels are very complex. Rule-based automatic enumeration parsing (*i.e.* splitting) turns out to be more difficult to adapt than the thesaurus used in a previous experiment [14] because of the various possible syntactic compositions. This could be explained by the fact that thesaurus items are well-designated concepts, whereas nomenclature items can be viewed as indications to guide the category choice.

For both methods, a better precision may be reached by weighting more efficiently extracted phrases according to the part of the document in which they occur (introduction, conclusion, past history and current illness description).

Finally, it will be interesting to conduct an evaluation of the help effectively provided by the tool to the coders.

As a conclusion, we can outline that our symbolic methods prove their usefulness in the context of unannotated corpora, where it is difficult to apply machine learning approaches. Moreover, we think they can successfully be mixed with learning algorithms when a training set is available.

Acknowledgements. This work was partly supported by the CAPADIS and STRATEGO projects. CAPADIS is funded by the government of the Brussels-Capital Region, Belgium (ISRIB). STRATEGO (WIST 2 project 616442) is funded by the government of Walloon Region, Belgium.

References

1. Ananiadou, S., McNaught, J.: Introduction to text mining in biology. In: Text Mining for Biology and Biomedicine, pp. 1–12. Artech House Books (2006)
2. Ceusters, W., Michel, C., Penson, D., Mauclet, E.: Semi-automated encoding of diagnoses and medical procedures combining ICD-9-CM with computational-linguistic tools. Ann. Med. Milit. Belg. 8(2), 53–58 (1994)
3. Zweigenbaum, P., Consortium Menelas: Menelas: Coding and information retrieval from natural language patient discharge summaries. In: Laires, M., Ladeira, M., Christensen, J. (eds.) Advances in Health Telematics, pp. 82–89. IOS Press, Amsterdam (1995)
4. Friedman, C., Shagina, L., Lussier, Y., Hripcsak, G.: Automated encoding of clinical documents based on natural language processing. Journal of the American Medical Informatics Association 11(5), 392–402 (2004)
5. Pakhomov, S.V., Buntrock, J.D., Chute, C.G.: Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. JAMIA 13(5), 516–525 (2006)
6. Pestian, J.P., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K.B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, ACL, Prague, Czech Republic, pp. 97–104 (2007)
7. Farkas, R., Szarvas, G.: Automatic construction of rule-based ICD-9-CM coding systems. BMC Bioinformatics 9(Suppl. 3), S10 (2008)
8. Goldstein, I., Arzrumtsyan, A., Uzuner, O.: Three approaches to automatic assignment of ICD-9-CM codes to radiology reports. In: Proceedings of AMIA Annual Symposium, pp. 279–283 (2007)
9. Pereira, S., Névéol, A., Massari, P., Joubert, M., Darmoni, S.: Construction of a semi-automated ICD-10 coding help system to optimize medical and economic coding. Studies in Health Technology and Informatics 124, 845–850 (2006)
10. Medori, J.: From free text to ICD: development of a coding help. In: Proc. of the 1st Louhi Workshop on Text and Data Mining of Health Documents, Turku (2008)
11. Paumier, S.: De la reconnaissance de formes linguistiques à l'analyse syntaxique. PhD thesis, Université de Marne-la-Vallée (2003)
12. Deléger, L., Namer, F., Zweigenbaum, P.: Morphosemantic parsing of medical compound words: transferring a french analyzer to english. International Journal of Medical Informatics 78(Suppl. 1), S48–S55 (2009)
13. Namer, F.: Automatiser l'analyse morpho-sémantique non affixale: le système DériF. Cahiers de grammaire 28, 31–48 (2003)
14. Kevers, L.: Indexation semi-automatique de textes: thésaurus et transducteurs. In: Actes de la 6e Conférence Francophone en Recherche d'Information et Applications, Presqu'île de Giens, France, pp. 151–167 (May 2009)
15. Kevers, L., Mantrach, A., Fairon, C., Bersini, H., Saerens, M.: Classification supervisée hybride par motifs lexicaux étendus et classificateurs SVM. In: Actes Des 10e Journées Internationales D'analyse Des Données Textuelles, Rome (June 2010)
16. Porter, M.F.: An algorithm for suffix stripping. In: Readings in Information Retrieval, pp. 313–316. Morgan Kaufmann Publishers Inc., San Francisco (1997)
17. Funk, M.E., Reid, C.A., McGoogan, L.S.: Indexing consistency in MEDLINE. Bulletin of the Medical Library Association 71(2), 176–183 (1983)

User-Tailored Document Planning – A Game-Theoretic Approach

Ralf Klabunde and Alexander Kornrumpf

Ruhr-Universität Bochum, Department of Linguistics, 44780 Bochum, Germany
{ralf.klabunde,alexander.kornrumpf}@rub.de

Abstract. In order to satisfy the informational demands of different users, generated texts should be tailored to the respective user types. Document planning may benefit from a formal modeling of the participating agents (the generation system and the user) within the framework of game theory. We show how rhetorical structures map to speaker strategies, and how a user model may be represented as a domain theory, containing different hypotheses for the listener strategies. Based on this, we present an algorithm which simultaneously performs the tasks of message selection and document structuring.

Keywords: natural language generation, document planning, game theory.

1 Introduction

During natural language generation (NLG), document planning is the first step in transforming non-linguistic informational units (database entries or concepts) into a coherent text. Document planning comprises two subtasks: the selection of the information to be conveyed from the underlying data or knowledge base, and merging the selected informational units to textual units. Typically, the latter subtask is performed by establishing rhetorical relations [6] between semantic representations for corresponding text spans. The result is a tree structure for the entire document plan.

Obviously, the variable information requirements of users result in different document plans with respect to content and structure. We aim at a general, game-theoretic model of document planning where different contents for various user types are determined by differing assumption costs and utilities for the players involved.

The close link between Gricean pragmatics and game theory is well-known since [5], but only recently game-theoretic modeling of information exchange became a flourishing area in pragmatics [7]. Document planning is driven by pragmatic demands since good guidelines for selecting the information to be conveyed are the Gricean maxims, ensuring effective communication [10]. Since almost every game-theoretic model of communication is an attempt to formalize the Gricean ideas, we achieve a precise reformulation of some of the Gricean ideas for document planning.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 209–214, 2010.

2 A Game-Theoretic Algorithm for Document Planning

Our starting point is a normal-form game (N, A, u) where N is a finite set of players, $A = A_1 \times \ldots \times A_n$ is the set of possible actions and $u : A \rightarrow \mathbb{R}^n$ is a utility function which denotes the payoff for every player, given a tupel of actions that is played [9]. The two players of our game are $\mathcal{S}$, the generation system or 'speaker', and $\mathcal{L}$, a user model that simulates the behaviour of the listener.

Formally, document planning is a function $m = \mathfrak{h}(\mathfrak{g}(d))$ where $\mathfrak{g} : d \mapsto m^*$ is a function that generates a set of messages m^* that can be derived from the knowledge source d and $\mathfrak{h} : m^* \mapsto m$ is a function which maps messages to a complex message with respect to the preferences of $\mathcal{S}$ and $\mathcal{L}$. In NLG terms, $\mathfrak{g}$ is responsible for content determination and $\mathfrak{h}$ determines the discourse relations.

2.1 Speaker and Listener Actions

We start with the specification of the actions A of the normal-form game (N, A, u).

Speaker actions. Despite the well-known weaknesses of an analysis based on rhetorical structures, we act on the assumption that rhetorical relations reflect the basic actions of $\mathcal{S}$ in order to select the messages and to structure the document.

Speaker actions play a major role in the discourse planning algorithm presented in subsection 2.3. This algorithm does not only realize $\mathfrak{h}$ but $\mathfrak{g}$ as well, because we map our data to message types. From the total amount of messages we receive, only those messages will be taken into consideration that are linked by rhetorical relations.

Listener actions. $\mathcal{L}$'s action set is closely linked to the preconditions and effects of the rhetorical relations used. Listener actions are responsible for the update of the information state in the user model. $\mathcal{L}$s interpretation task is to find an explanation for the information on the basis of his own beliefs. Since interpretation is equivalent to finding an explanation why the information conveyed might be true, weighted – or cost-based – abductive reasoning is the suitable mechanism for the update of the information state. We use the notion of a domain theory [1] to define $\mathcal{L}$'s beliefs about a domain:

Definition 1. ***Domain Theory:*** *An agent $\mathcal{A}$'s knowledge and beliefs about a specific domain D is called a domain theory $T_{\mathcal{A}}(D)$ iff:*

1. *$T_{\mathcal{A}}(D)$ is a partially ordered set*
2. *$T_{\mathcal{A}}(D)$ is a propositional logic program, i.e. each $t \in T_{\mathcal{A}}(D)$ is of the form $\alpha_1 \wedge \ldots \wedge \alpha_i \wedge \neg\beta_1 \wedge \ldots \neg \wedge \beta_j \rightarrow p = body \rightarrow head$.*

In our approach, the head of a formula is either a message of a certain type, or it represents a rhetorical relation. In this case, the body describes a precondition of that relation. A domain theory does not only encode user knowledge and beliefs, but it is particularly important for the representation of the reasoning capabilities of $\mathcal{L}$. They form the core concept for weighted abduction:

Definition 2. ***Candidate hypothesis:*** *Be $T_{\mathcal{A}}$ a domain theory and P the set of predicate symbols within $T_{\mathcal{A}}$. Then $H = \{p \in P : p$ does not occur in a head clause of $T_{\mathcal{A}}\}$ is the set of abducibles for $T_{\mathcal{A}}$ and $\overline{H} = P \setminus H$ the set of non-abducibles. $h \in H$ is called a candidate hypothesis.*

Definition 3. ***Abduction:*** *Be $T_{\mathcal{A}}$ a domain theory and $\Psi : \Psi \cap H = \emptyset$ a set of observations to be explained. Find a consistent explanation formula F with $F \Rightarrow \Psi \wedge (F \cap \overline{H} = \emptyset)$. In other words, abduction means: given a set Ψ of non-abducibles, find a formula F which consists solely of abducibles and explains Ψ.*

Since there is usually more than one explanation F which explains Ψ, external criteria are needed to determine the best explanation. This can be done by assigning costs to each candidate hypothesis h. Abduction then becomes an optimization problem, usually called weighted abduction.

Definition 4. ***Weighted or cost-based abduction:*** *Be $(T_{\mathcal{A}}, \Psi)$ an abduction problem. Minimize $\sum_{h \in F} costs(h)$ subject to $F \Rightarrow \Psi \wedge (F \cap \overline{H} = \emptyset)$.*

Given a generated document plan, the action set of $\mathcal{L}$ comprises all updates of his information state which explain why the document plan might be true.

2.2 The Utility Function

The central concept for every game-theoretic model is the utility function. All aspects we have mentioned so far lead to plausible utilities or payoffs for $\mathcal{S}$ and $\mathcal{L}$, respectively. The utility function is not only the core concept, the exact fixing of the payoffs is also the most problematic decision for every game. While in economic scenarios the payoffs are often associated with monetary values, in our approach the payoffs represent the cognitive burden of the agents, which is hard to quantify. However, the exact numerical utilities are not the crucial factor but the relation between the different payoffs. Therefore, we do not postulate that the exact numerical utilities bear any deep semantics. The differences between the payoffs determine the preferences of the players and their best response to the actions of the other agent.

Let us assume that $\mathcal{S}$ has a set of document plans at its disposal which express the same data in different manners. Furthermore, $\mathcal{L}$ knows a set of hypotheses $A_{\mathcal{L}} = H$ which offer possible explanations for the data. H may be computed from $\mathcal{L}$'s domain theory $T_{\mathcal{L}}$. Communication between $\mathcal{S}$ and $\mathcal{L}$ requires that $\mathcal{S}$ chooses a rhetorical relation $a_{\mathcal{S}} \in A_{\mathcal{S}}$ and conveys the relation and its arguments to $\mathcal{L}$ who in turn chooses a hypothesis $a_{\mathcal{L}} \in A_{\mathcal{L}}$ as an interpretation. The utility function provides a basis for the agents to make their choices.

Payoffs for $\mathcal{L}$. Cost-based abduction actually requires only very little reformulation to fit into the framework of game theory, since the notion of costs, i.e. negative utility, is already accounted for in that concept. For reasons of computational feasibility, in our approach $\mathcal{L}$ may only adhere to a single hypothesis which does not have to match all of the facts. Hence, in addition to costs we

need a metric of how good a hypothesis fits into the observed facts. We call the selection of a hypothesis on this basis *naive abduction*. $\mathcal{L}$'s utility may be formulated in a similar way:

Definition 5. ***Naive abduction:*** *Be $(T_{\mathcal{A}}, \Psi, costs)$ a cost-based abduction problem. Find $h^* \in H$ such that $\forall h \in H : match_{T_{\mathcal{A}}}(h, \Psi) - costs(h) \leq match_{T_{\mathcal{A}}}(h^*, \Psi) - costs(h^*)$.*

Definition 6. ***Listener utility:*** *Be $(a_{\mathcal{S}} = m, a_{\mathcal{L}} = h)$ a strategy profile with the interpretation as given above. Let $T_{\mathcal{L}}$ be the domain theory of $\mathcal{L}$ and $\Psi(m)$ the propositions covered by $a_{\mathcal{S}}$. The utility of $\mathcal{L}$ is defined as follows: $u_{\mathcal{L}}(m, h) := \alpha_1 \cdot match_{T_{\mathcal{L}}}(h, \Psi(m)) - \alpha_2 \cdot costs(h)$.*

$\alpha_{1,2}$ are positive coefficients that formalize the priorities of $\mathcal{L}$. In order to completely flesh out the definition, the parameters *costs*, *match* and $\Psi(m)$ must be defined. The values of these parameters depend to some extend on the domain, so that we will outline the basic requirements for their complete definition.

As for *costs*, its definition does not depend on $a_{\mathcal{S}}$. Each hypothesis h will be assigned a scalar value which indicates how likely $\mathcal{L}$ is to believe h. On an ordinal scale this can be derived from considerations about the user type.

For *match* we use the following domain-independent auxiliary definitions:

$$match_{T_{\mathcal{A}}}(h, \Psi) = \sum_{p : h \Rightarrow p} \sum_{q \in \Psi} match'_{T_{\mathcal{A}}}(p, q)$$

$$match'_{T_{\mathcal{A}}}(p, q) = \begin{cases} \gamma_1 \text{ if } q \rightarrow p \\ -\gamma_2 \text{ if } \neg q \rightarrow p \\ 0 \text{ otherwise} \end{cases}$$

By means of this definition, the matching problem is reduced to comparing the propositions p which can be derived from h with the propositions $q \in \Psi$. In addition, $\Psi(m)$ does not only decode the messages, but also the rhetorical structure within m.

Payoffs for $\mathcal{S}$. The definition of $\mathcal{S}$'s utility relies on the common assumption that the production of longer or more complex expressions is penalized in some way (see, e.g., [2]) . Such a penalty for more complex productions forces $\mathcal{S}$ to consider only those data in d that are relevant to $\mathcal{L}$:

Definition 7. ***Irrelevant data:*** *Be $\mathfrak{a}$ the naive abduction function and $d \in D$ a set of informational units. A subset $\overline{d} \subseteq D$ is called irrelevant iff $\mathfrak{a}(T_{\mathcal{A}}, d \setminus \overline{d}, costs) = \mathfrak{a}(T_{\mathcal{A}}, d, costs)$. That is, the agent arrives at the same hypothesis regardless of the knowledge of $\overline{d}$.*

Definition 8. ***Speaker utility:*** *Be $(a_{\mathcal{S}} = m, a_{\mathcal{L}} = h)$ a strategy profile as given above. $T_{\mathcal{L}}$ is the domain theory of $\mathcal{L}$, and $complexity(m) = |nodes \in m|$ a metric for the complexity of m. The utility of $\mathcal{S}$ is defined as: $u_{\mathcal{S}}(m, h) := \frac{\beta_1 \cdot match_{T_{\mathcal{L}}}(h, d)}{\beta_2 \cdot complexity(m)}$.*

The utility function for $\mathcal{S}$ has to find a balance between the goal of leading $\mathcal{L}$ to a hypothesis that accounts for d and minimizing the complexity of the communicated document plan m.

2.3 A Multi-iteration Game Algorithm for Discourse Planning

We are now able to achieve text plans with our game-theoretic concepts. As already mentioned, our task is to compute $\mathfrak{h}(\mathfrak{g}(d))$. The algorithm given in Table 1 determines the relevant subset of all possible messages and the structure of the document plan simultaneously: The algorithm combines only those messages by means of rhetorical relations that form, together with a listener's update mechanisms, a Nash-equilibrium.

Table 1. A game-theoretic document planning algorithm

```
1: POOL ← all messages derived from d
2: N ← {S, L}
3: A_L ← H
4: u ← (u_S, u_S)
5: R ← nil
6: repeat
7:     A_S ← speaker actions: {rhet. relations which may link pairs of elements in
           POOL} ∪ POOL
8:     (m, h) ← pure strategy equilibrium of (N, A, u)
9:     if m ∈ POOL then
10:        R ← m
11:    else
12:        E ← {constituents of a_S}
13:        POOL ← POOL\E
14:        POOL ← POOL ∪ m
15:    end if
16: until R ≠ nil
17: return R
```

There are some important differences between this algorithm and the algorithm presented in [8, p. 108]. Therefore, we will explain this algorithm in some detail.

Line 1: Instead of preselecting messages by means of heuristics, the pool is initialized with all messages known to $\mathcal{S}$.

Lines 2–4: Initialize the invariant parts of the game according to the definitions given above. While the set of hypotheses in $\mathcal{L}$'s domain theory never changes, $\mathcal{S}$'s available actions change in every iteration.

Lines 5–6, 16–17: Since the algorithm may return a document plan that covers only a subset of the message pool, it performs content determination as well.

Line 8: The standard solution concept in game theory is the Nash-equilibrium, i.e. (m, h) are mutually best responses.

Lines 9-10: If $\mathcal{S}$ is in an equilibrium and realizes an existing rhetorical relation instead of constructing a new one, the pool does not change.

Lines 11–15: If $\mathcal{S}$ constructs a new message combination by means of a rhetorical relation, the pool is updated and the loop starts over.

3 Summary and Outlook

To sum up, we provided generic formal notations for the interlocutors, their tasks, action sets and the utility functions. All definitions are grounded in well-known theoretical frameworks, and game theory allowed the formulation of the interplay of the relevant representations and processes rooted in these theories.

In addition to developing the formal model of document planning mentioned above, we applied that model to the generation of document plans for different users of performance data. The generated texts explain the output of a heart-rate monitor (HRM) worn by a runner during his training for amateur atheletes and beginners. For reasons of space, we cannot go into the details; [4] will provide a comprehensive overview.

References

1. Console, L., Dupre, D., Torasso, P.: On the relationship between abduction and deduction. Journal of Logic and Computation 1(5), 661–690 (1991)
2. Jäger, G.: Evolutionary game theory and typology: a case study. Language 83, 74–109 (2007)
3. Klabunde, R.: Towards a game-theoretic approach to document planning. In: Proceedings of ENLG 2009, pp. 102–105 (2009)
4. Klabunde, R., Kornrumpf, A.: A game-theoretic approach to document planning for performance data (forthcoming)
5. Lewis, D.: Convention. Harvard University Press, Cambridge (1969)
6. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organisation. Text 8(3), 243–281 (1988)
7. Pietarinen, A.-V. (ed.): Game Theory and Linguistic Meaning. Elsevier, Amsterdam (2007)
8. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press, Cambridge (2000)
9. Shoham, Y., Leyton-Brown, K.: Multiagent Systems. Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, Cambridge (2009)
10. Sripada, S.G., Reiter, E., Hunter, J., Yu, J.: Generating English Summaries of Time Series Data Using the Gricean Maxims. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 198–196 (2003)

Anaphora Resolution with Real Preprocessing

Manfred Klenner, Don Tuggener, Angela Fahrni, and Rico Sennrich

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

Abstract. In this paper we focus on anaphora resolution for German, a highly inflected language which also allows for closed form compounds (i.e. compounds without spaces). Especially, we describe a system that only uses real preprocessing components, e.g. a dependency parser, a two-level morphological analyser etc. We trace the performance drop occurring under these conditions back to underspecification and ambiguity at the morphological level. A demanding subtask of anaphora resolution are the so-called bridging anaphora, a special variant of nominal anaphora where the heads of the coreferent noun phrases do not match. We experiment with two different resources in order to find out how to cope best with this problem.

Keywords: Anaphora Resolution, Coreference Resolution, Bridging Anaphora.

1 Introduction

Anaphora resolution is a resource-intensive task. In order to find out whether a noun phrase is an antecedent of another (subsequent) noun phrase, the anaphor, information from various preprocessing components are to be combined. A morphological analyser is needed for number, person and gender determination, a tagger is required to deliver part of speech tags, a parser to find grammatical functions and the embedding depth of noun phrases and finally semantic information is necessary to tackle the most difficult task, namely, bridging anaphora. Bridging anaphora are nominal anaphora where the heads of the noun phrases do not match. Take the following sequence: 'Iceland is an interesting place to visit. The land of ice and fire is famous for ....' Here, 'Iceland' and 'land of ice and fire' are coreferent. In order to establish this coreference link, the least a system has to know is that Iceland is a land. Lexical resources such a WordNet or its German counterpart GermaNet do comprise this kind of information although not exhaustively. The proper determination of coreference depends on the quality of these resources and the preprocessing units using them. Thus, a poor performance of a system for anaphora resolution can have multiple causes and often it is hard to tell which component or resource is to blame. Therefore, it is tempting to reduce this kind of noise to its minimum and to create idealised

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 215–225, 2010.

conditions under which one can easily fix failures. Instead of using a parser one could use a treebank and if the treebank also has morphological annotations why not use it as well. This way, one ends up with a system that expects perfect preprocessing and whose empirical results no longer indicate its usefulness for real-world applications. This kind of simplifications are often made by current approaches to anaphora resolution. One of the most unrealistic and simplifying idealisations is to use true mentions instead of all noun phrases. True mentions are those markables that are - according to a coreference gold standard - part of a coreference chain. The majority of noun phrases in a text, however, are not in a coreference set. The determination whether a NP is anaphoric (i.e. a true mention) or not is a demanding problem, the so called anaphoricity classification problem. There are a few systems that incorporate anaphoricity classification, the majority of systems leaves this as an implicit task to the anaphora resolution component. Separate anaphoricity classification has not proven to be more successful than its implicit counterpart. Anaphoricity determination of markables is a non-trival task and cutting it away makes a system an artificial one.

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

In this article we introduce a system for anaphora resolution for German that uses only real preprocessing components: Gertwol, a morphological analyser; Pro3Gres, a dependency parser; GermaNet, a German wordnet and Wortschatz Leipzig, a lexical resource generated by statistical means. As most approaches, we cast anaphora resolution as pairwise classification - we use TiMBL (Daelemans et al., 2004) as a machine learning tool. Our system is filter-based that is, candidate pairs that do not fulfil linguistic filter criteria are sorted out. We give empirical results and discuss the reason for the drop of performance from an idealised setting to a real-world setting. Also, different filters have been investigated to determine the usefulness of lexical resources for the task of resolving bridging anaphora for German.

2 Filter-Based Pairwise Classification

Approaches to pairwise classification of anaphora resolution differ, among others, in their pair generation module. Some systems generate every pair independent of the distance between two *markables* (the noun phrases that might stand in a coreference relation). Under a linguistic point of view this only makes sense for nominal anaphora. A pronoun at the end of a text could hardly refer back to a noun phrase at the beginning of a text without further intervening chain links. Moreover, the problem with such an approach is the vast amount of negative instances it produces – the learned classifier gets biased towards negative classification. The number of negative pairs is a problem anyway, even in systems which work with a fixed window within pairs are searched. Soon et al., 2001 use a dynamic window (for training only), where all pairs are generated until the real

antecedent of an anaphor is reached. We use a fixed window of three sentences for pronominal anaphora and bridging anaphora, while for named entities there is no restriction. Each pair additionally must pass all applicable filters. Filters depend on the part of speech of the antecedent-anaphor candidate. For instance, personal pronouns must agree in person, number and gender with its antecedent head (whether this is a pronoun or a noun). After morphological analysis, we often have underspecified information at hand only. For instance, German 'ihr' can be plural without gender restriction ('their') or singular feminine ('her'). If no information is available (e.g. for unknown nouns) we take a disjunction of all allowed values. Possessive pronouns only unify in person and gender, e.g. 'Sie liebt ihre Bücher' ('She^i loves her^i books'), but not in number. 'ihre' ('her') is plural, 'Sie' ('She') is singular. Nominal anaphora in German must only agree in number (and trivially in person), but not necessarily in gender ('Der Weg$^i_{masc}$ ist lang. Ich bin diese Strecke$^i_{fem}$ ...'). Each of these cases is covered by a rule and there are some rules for special cases, e.g. the rule for reported speech, where a third person pronoun is coreferent with a first person pronoun, e.g. 'Er^i sagte: "ich^i ..."' ('He^i said: "I^i ..."').

Besides the morphological, there are syntactic and semantic filters. Among the syntactic filters, the subclause filter is the most prominent. It can be used to operationalize binding constraints and helps to reduce the amount of negative pairs. The constraint here is: two personal pronouns (or nouns) in the same subclause cannot be coreferent ('Sie^i vertraut ihr^j', where $i \neq j$; 'She^i trusts her^j'). With possessive pronouns this is different, a possessive pronoun and its antecedent are allowed to co-occur in the same subclause. For reflexive pronouns the antecedent even should be in the same subclause, but there are exceptions (sentences where the reflexive pronoun is not anaphoric at all).

Semantic filters are based on GermaNet (Hamp and Feldweg, 1997), the German wordnet and Wortschatz Leipzig (http://www.wortschatz.uni-leipzig.de). Two nominal markables must be semantically compatible, which means that they must be both e.g. animate or inanimate, or stand in a hyponym or synonym relation. See section 4 for our experiments with bridging anaphora.

We strive to integrate as much linguistic knowledge as possible into the filters. Alternatively, one could use this kind of linguistic knowledge as a feature. But our experiments have shown that a filter based approach is more reliable. There are only a few exceptions of these regularities (at least at the morphological and syntactic level). It is better to erroneously filter out such pairs as to let everything pass. But of course, underspecified or uncertain information as produced by real components is a problem. We evaluate the performance drop when relaxing gold standard information in section 4.

A pair that has passed all filters is given to the classifier. Except of salience, we do not introduce new features in our approach, instead, we use standard features found in the literature (e.g. Soon et al., 2001). We work with the memory-based learner TiMBL (Daelemans et al., 2004) as a machine learning classifier.

Here is the list of our features:

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

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

It has been noticed that the local perspective of pairwise classification yields problems. Take the following markable chain: 'Hillary Clinton ... she ... Angela Merkel'. 'she' is compatible with 'Hillary Clinton', 'Angela Merkel' is compatible with 'she', but 'Merkel' and 'Clinton' are incompatible. Since transitivity is outside the scope of a pairwise classifier, it might well classify both compatible pairs as positive without noticing that this leads to an implicit contradiction (setting 'Clinton' and 'Merkel' to be coreferent). In a former paper we have argued that coreference clustering based on the so-called Balas order coupled with intensional constraints to ensure consistency of coreference sets performs best in order to remedy these problems (Klenner and Ailloud, 2009). In this paper, we concentrate on the performance drop of the baseline system under the conditions of real preprocessing components. We do not discuss problems of coreference clustering.

3 Real Preprocessing Tools

Fortunately, good NLP tools are available for a number of languages. For German, a two-level morphology program called Gertwol, a fast and well performing part-of-speech tagger, the TreeTagger (Schmid, 1994), and a fast and state-of-the-art dependency parser, the Pro3Gres parser, are the components of our systems. Additionally, we have developed a named-entity recognition based on pattern matching and Wikipedia entries. It is evident that the quality of preprocessing determines the quality of the rest, namely, the decision made by linguistic filters and the classification carried out by the machine learning classifier.

3.1 Morphology with Gertwol

We use Gertwol, a commercial system based on two-level morphology. Gertwol is fast and also carries out noun decomposition which is rather useful, since in

German compounds are realised as single wordforms (closed form compounds), e.g. Computerexperte ('computer expert'). Compounds (which are quite frequent in German) might become very complex, but often the head of the compound is sufficient to semantically classify the whole compound via GermaNet. For instance, 'Netzwerkcomputerexperte' ('expert for network computers') is an expert and, thus, is animate. The other important task of Gertwol is to determine number, person and gender information of a word. Unfortunately, ambiguity rate is high, since e.g. some personal pronouns are highly ambiguous. For instance, the German pronoun 'sie' ('she') might be singular/feminine or plural (without gender restriction). The pronoun 'ich' does not impose any gender restrictions and moreover often refers in reported speech to a speaker which is referred to in the text by a noun phrase in third person.

3.2 Named-Entity Recognition

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

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

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

3.3 Pro3gresDe: The Parser

Pro3GresDe is a hybrid dependency parser for German that is based on the English Pro3Gres parser (cf. Schneider, 2008). It combines a hand-written grammar and a statistical disambiguation module trained on part of the TüBa-D/Z treebank (see Telljohann et al., 2004).[1]. This hybrid approach has proven especially useful for the functional disambiguation of German noun phrases. While the function of noun phrases is marked morphologically in German, many noun phrases are morphologically ambiguous, especially named entities. We use both morphological unification rules and statistical information from TüBa-D/Z (i.e. data about possible subcategorisation frames of verbs) to resolve functional ambiguities. We have shown that this approach performs better at functionally disambiguating noun phrases than purely statistical parsers.

[1] For a full discussion of Pro3GresDe, see Sennrich et al., 2009.

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

4 Empirical Evaluation

We have carried out two series of experiments. The first one is concerned with the costs of real preprocessing compared to the use of gold standard information (e.g. tree bank instead of parser). We incrementally fix the reasons for the performance drop. The second experiments are devoted to bridging anaphora and the impact of two main lexical resources for German: GermaNet, a German WordNet and Wortschatz Leipzig, a statistically derived thesaurus.

4.1 The Price of Real Preprocessing

From the two processing steps of coreference resolution - pairwise classification and subsequent clustering - only the first is of interest here. It is the baseline performance drop that we are interested in. This degradation occurs before clustering and it cannot be compensated by clustering operations.

The performance drop is measured in terms of save (gold standard) versus noisy (real-world components) morphological, functional and syntactic information. The gold standard information stems from the TüBa-D/Z treebank (phrase structure trees, grammatical functions, head information and morphology) which also is annotated with coreference links (Naumann, 2006). Our experiments are restricted to nominal anaphora and personal pronouns, i.e. we exclude the simple cases of reflexive and relative pronouns, but also possessive pronouns, since we are focusing on the most demanding classes.

In a first step, we have run the system with all markables and without any gold standard information (see Tab. 1). The f-measure of these runs (5-fold cross validation) is 58.01%, with a precision of 70.89% and a recall of 49.01%. The performance is low because recall is low; precision is good. Recall is low for different reasons. First of all, our filters for nominal anaphora are quite restrictive (fuzzy string matching, GermaNet hyponym and synonym restrictions). Many of the false negatives stem from such filtered out nominal pairs. Refining our filters for nominal anaphora could help to improve recall. Some of our experiments concerning bridging anaphora are described in the next section.

A different reason for low recall is the fixed window of 3 sentences. Only named-entities are allowed to refer back further than 3 sentences, but not personal pronouns and common nouns. This way, we miss some long distance

Table 1. Performance Drop

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

anaphoric relations. Our experiments have, however, shown that it is better to restrict the search than to generate each and every pair: performance drops to a great extent the larger the window.

Finally, the local perspective of pairwise classification does not allow to take boundness restrictions into account. For instance, we know that third person personal pronouns (and possessive pronouns as well) are anaphoric (i.e. must be bound) - there are only very few exceptions. There is, however, no way to tell the learner this kind of prior knowledge. Fortunately, this shortcoming can be compensated at the subsequent clustering step, where these markables can be forced to be bound to the best available anaphor.

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

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

We were quite surprised that the replacement of syntactic and functional information by real components was not the problem. Morphology is responsible for the drop.

4.2 Filtering for Resolution of Bridging Anaphora

In this section we show that, using different morpho-syntactic, distance-based and semantic filters derived from real resources, the task of resolving bridging anaphora in a pairwise manner is far from being accomplished with satisfying results. Filtering aims at reducing the number of negative instances, but this has been hardly investigated regarding the ceiling or performance upper bound it produces. The upper bound values given in Tab.2 indicate how many false negatives a filter produces (i.e. how many real positives it filters out)[2]. We have further investigated these upper bounds (see Tab. 3) and found that they are either very low when using very restrictive ('strict') filters or that the filters do not eliminate enough negative instances when used in a relaxed ('lax') mode. Throughout our experiments, we use the CEAF scorer presented in Luo, 2005.

[2] We get a slight reduction in precision when using no filters because of a string matching issue when filtering out string matching multiword items.

Table 2. Upper Bound of the Morpho-syntactic and Distance Filters

Filter	Recall	Precision	F-measure	Pairs	Reduction	Positives
no filter	100.00	98.60	99.21	4869822	-	4924 (0.10%)
diff_regens	99.87	98.53	99.11	4864018	-0.10%	4915 (0.10%)
anaphor_definite	100.00	98.60	99.21	4401565	-9.62%	4913 (0.11%)
number_agreement	93.91	94.64	94.00	3480538	-28.53%	4622 (0.13%)
all_morphosynt_filters	93.78	94.57	93.90	3110842	-36.12%	4602 (0.15%)
dist limit 3	68.36	80.54	72.46	818588	-83.19%	1697 (0.21%)
all	63.31	76.82	67.81	520735	-89.30%	1579 (0.30%)

We can see from Tab. 2 that the morpho-syntactic filters, which perform well in resolving pronominal anaphora, give good upper bounds but do not reduce the amount of negative instances sufficiently. Subclause exclusion (here diff_regens: determined through verb dependency), which establishes a kind of c-command in our dependency framework, is not really that relevant for resolving bridging anaphora, as antecedents are often not in the same sentence. Perhaps surprising is the fact that 9 positive instances get deleted by this filter. Such errors occur with real preprocessing, as parsing is not perfect. A simple definiteness filter (anaphor_definite) that checks if a candidate anaphor has an indefinite determiner (German "ein", i.e. 'a or an'; or its morphological variants) reduces the training instances by almost 10% without reducing the upper bound. Number agreement filtering shows that there are 302 positive instances that do not agree in number. Still this filter cuts down the number of instances by almost 30%. The often used distance filter with a sentence window of 3 produces an acceptable upper bound and reduces the instance size by 83.91%. This is still not enough, however, looking at the percentage of positives (0.21%).

For filtering based on semantic information we use Wortschatz Leipzig and GermaNet. We apply head extraction and decomposition to composite nouns based on Gertwol morphological analysis, in the case they are not found directly in the lexical resources.

For 54'593 (83,1%) of the 65'703 markables synonyms can be found in Wortschatz Leipzig (WSL), for 60'985 (92,8%) we can make a (often ambiguous) GermaNet (GN) classification. The synonymy filter WSL checks if a mention is in the synonymy list of the other one or if they share a common synonym. The GN filter checks if both mentions are in the same GN class (if the class is ambiguous we check all and let the pair be generated if we find a match). We investigate the upper bounds of the semantic filters in two ways (see Tab. 3): If for a mention no information has been derived, we let it pass the filter ('lax') or we delete it ('strict').

There are huge differences between the upper bounds and the percentages of positive instances between' lax' and 'strict' filtering. This suggests that although for quite a large number of markables semantic information can be retrieved, it does not allow us to use it for hard filtering without a significant drop in the upper bound ceiling. This gets obvious when we combine the strict semantic

Table 3. Upper Bound of the Semantic Filters

Filter	Recall	Precision	F-measure	Pairs	Reduction	Positives
WSL (strict)	37.00	55.71	42.34	112921	-97.86%	1590 (1.41%)
WSL (lax)	72.94	83.35	76.38	1679610	-65.51%	3300 (0.20%)
GN (strict)	57.98	73.52	63.21	1030441	-78.84%	3283 (0.32%)
GN (lax)	81.36	89.03	84.04	1694157	-65.21%	4385 (0.26%)
all_filt (lax)	36.86	53.97	41.93	97593	-98.00%	1013 (1.04%)
all_filt (strict)	15.1	28.41	18.32	8953	-99.81%	441 (4.93%)

constraints with the morpho-syntactic and distance filters in all_filt (strict). It is the only filter that generates a fairly reasonable percentage of positives, but drops the upper bound immensely.

As one would expect, a synonymy based filter (WSL) is more strict than a semantic class based constraint (GN). The trade-off between the percentage of positives and the reduction of the upper bound holds equally for both of the semantic filters: the more positives, the lower the upper bound.

Filtering is a key element to successful resolution of bridging anaphora. However, our experiments show that filters based on morphological information and syntactical constraints do not sufficiently reduce the amount of negative instances in order to train a reasonable classifier. The distance constraint is a good filter to tune the trade-off between recall and precision, although the distance values might be highly dependent on the test domain and genre. On the other hand, using the lexical resources for filtering based on semantic constraints heavily suffers from sparseness, leading to a considerably lower upper bound.

These findings seem to suggest that pairwise classification is not the best technique for resolving bridging anaphora given a real anaphora resolution scenario. We are currently carrying out experiments with an incremental approach, where pairwise classification is done only between the last mentions of already established coreference sets and the anaphor candidate. We hope to show that by recasting the problem of coreference resolution as an incremental clustering problem the issue of resolving bridging anaphora becomes less important - because true mentions linked through a bridging relation can be merged by a pronoun between them.

5 Related Work

The work of Soon et al., 2001 is a prototypical and often reimplemented machine-learning approach in the paradigm of pair-wise classification. Our system has a similar baseline architecture, and our features do overlap to a great extent.

Work on coreference resolution for German is rare, most of it uses the coreference annotated treebank TüBa-D/Z. Versley, 2006 uses a maximum entropy model for nominal anaphora resolution, his major insight is that if information from GermaNet is available then it outperforms the statistical model. We

took this finding seriously and have tried to use Wikipedia to complement GermaNet (we map Wikipedia multiword items via Wikipedia categories to GermaNet classes). We also have experimented with a statistically derived lexical resource, the Wortschatz Leipzig.

Hinrichs et al., 2005 introduce anaphora resolution (only pronouns) on the basis of a former version of the TüBa-D/Z. They also work with TiMBL. Their results are based on treebank gold standard information and are – compared to subsequent work, cf. Wunsch et al., 2009, where also gold standard information was utilised – surprisingly high (f-measure 73.40% compared to 58.40%).

A study concerning the influence of different knowledge sources and preprocessing components on pronoun resolution was carried out by Schiehlen, 2004. A gold standard created by the author was used for evaluation (based on the Negra corpus).

Klenner and Ailloud, 2008 and Klenner and Ailloud, 2009 are concerned with the consistency of coreference sets using idealised input from the TüBa-D/Z treebank.

6 Conclusion

In this paper, we have discussed the intricacies of anaphora resolution based on real preprocessing components. Our system makes extensive use of non-statistical resources (rule-based dependency parsing, a German wordnet, Wikipedia, two-level morphology) but at the same time is based on a state of the art machine learning approach. We have traced the performance drop that occurs under this conditions back to its origin. It is the morphology of German that yields the problem. Although German counts as a highly inflected language, underspecification and ambiguity prevail and are the main cause of degrading performance.

We have also evaluated the usefulness of two resources, GermaNet and Wortschatz Leipzig. Our experiments suggest that filtering for pairwise classification is not a successful technique if bridging anaphora are concerned. Other methods for finding proper antecedent-anaphor candidates are needed here. Our initial experiments with an incremental model are promising, our future work will proceed in this direction.

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

References

[Daelemans et al., 2004]Daelemans, W., Zavrel, J., van der Sloot, K., van den Bosch, A.: TiMBL: Tilburg Memory-Based Learner (2004)

[Hamp and Feldweg, 1997]Hamp, B., Feldweg, H.: GermaNet—a Lexical-Semantic Net for German. In: Proc. of ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications (1997)

[Hinrichs et al., 2005]Hinrichs, E., Filippova, K., Wunsch, H.: A Data-driven Approach to Pronominal Anaphora Resolution in German. In: Proc. of RANLP, Borovets, Bulgaria (2005)

[Klenner and Ailloud, 2008]Klenner, M., Ailloud, E.: Enhancing Coreference Clustering. In: Johansson, C. (ed.) Proc. of the Second Workshop on Anaphora Resolution (WAR II), Bergen, Norway. NEALT Proceedings Series, vol. 2 (2008)

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

[Luo, 2005]Luo, X.: On coreference resolution performance metrics. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics Morristown, NJ, USA, pp. 25–32 (2005)

[Naumann, 2006]Naumann, K.: Manual for the Annotation of In-document Referential Relations. Electronic document (2006), http://www.sfs.uni-tuebingen.de/~de_tuebadz.shtml

[Schiehlen, 2004]Schiehlen, M.: Optimizing algorithms for pronoun resolution. In: Proc. of the 20th International Conference on Computational Linguistics (2004)

[Schmid, 1994]Schmid, H.: Probabilistic Part–of–Speech Tagging Using Decision Trees. In: Proc. of the Conference on New Methods in Language Processing, Manchester, UK (1994)

[Schneider, 2008]Schneider, G.: Hybrid Long-Distance Functional Dependency Parsing. Doctoral Thesis, Institute of Computational Linguistics, Univ. of Zurich. (2008)

[Sennrich et al., 2009]Sennrich, R., Schneider, G., Volk, M., Warin, M.: A New Hybrid Dependency Parser for German. In: Proc. of the German Society for Computational Linguistics and Language Technology 2009 (GSCL 2009), Potsdam, Germany, pp. 115–124 (2009)

[Soon et al., 2001]Soon, W., Ng, H., Lim, D.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics 27(4), 521–544 (2001)

[Telljohann et al., 2004]Telljohann, H., Hinrichs, E.W., Kübler, S.: The TüBa-D/Z Treebank: Annotating German with a Context-Free Backbone. In: Proc. of the Fourth Intern. Conf. on Language Resources and Evaluation, Lisbon, Portugal (2004)

[Versley, 2006]Versley, Y.: A Constraint-based Approach to Noun Phrase Coreference Resolution in German Newspaper Text. In: Konferenz zur Verarbeitung Natürlicher Sprache, KONVENS (2006)

[Wunsch et al., 2009]Wunsch, H., Kübler, S., Cantrell, R.: Instance Sampling Methods for Pronoun Resolution. In: Proc. of RANLP, Borovets, Bulgaria (2009)

Automatic Construction of a Morphological Dictionary of Multi-Word Units

Cvetana Krstev[1], Ranka Stanković[2], Ivan Obradović[2], Duško Vitas[3], and Miloš Utvić[1]

[1] Faculty of Philology, University of Belgrade
[2] Faculty of Mining and Geology, University of Belgrade
[3] Faculty of Mathematics, University of Belgrade

Abstract. The development of a comprehensive morphological dictionary of multi-word units for Serbian is a very demanding task, due to the complexity of Serbian morphology. Manual production of such a dictionary proved to be extremely time-consuming. In this paper we present a procedure that automatically produces dictionary lemmas for a given list of multi-word units. To accomplish this task the procedure relies on data in e-dictionaries of Serbian simple words, which are already well developed. We also offer an evaluation of the proposed procedure on several different sets of data. Finally, we discuss some implementation issues and present how the same procedure is used for other languages.

Keywords: electronic dictionary, Serbian, morphology, inflection, multi-word units, noun phrases, query expansion.

1 Introduction

We have been developing morphological electronic dictionaries of Serbian for natural language processing for many years now. Our e-dictionaries follow the methodology and format known as DELAS/DELAF, which is presented for French in [1][1]. Serbian e-dictionaries of simple forms have reached a considerable size: they have a total of 122,000 entries [2]. Although we are continually enlarging our e-dictionaries of simple words, we have taken a further step towards tackling the problem of multi-word units (MWUs). In an introduction in [3], supported by comprehensive references, Savary states that MWUs are hard to define controversial linguistic objects. Nevertheless, it seems that many authors agree that MWUs: (a) are composed of two or more graphical words; (b) show some degree of morphological, syntactic, distributional, or semantic non-compositionality; and (c) have unique and constant references. A deeper discussion of the nature of MWUs is beyond the scope of our paper. When dealing with MWUs we are taking a pragmatic approach and consider MWUs to be sequences of graphical units that for some reason have to be described and processed as a unit.

[1] A comprehensive bibliography on e-dictionaries for other languages developed using the same methodology is given at http://www-igm.univ-mlv.fr/~unitex/

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 226–237, 2010.

For some productive classes of MWUs, like different types of numerals and named entities (time and duration, measures and currencies), we have developed finite-state transducers (FSTs) that rely on morphological e-dictionaries of simple words to model these MWUs correctly [4]. When applied to a text in automatic text analysis these FSTs associate recognized MWUs with lemmas as well as with appropriate grammatical categories. Both the associated lemmas and grammatical categories are in the same format as the one provided by dictionaries of simple words.

Some other MWUs, that are idiosyncratic in nature, ask for a different description. Namely, these MWUs cannot be described by FSTs - they have to be listed in an e-dictionary in a similar way and for similar reasons as simple words. That means that some regular form, or a lemma, has to be listed in a DELAC dictionary, together with some additional information that would enable the generation of all inflected forms, that is, entries for a DELACF dictionary. Several questions arise here: (a) what should this regular form listed in a dictionary of lemmas be; (b) what additional information is necessary; and (c) how the generation of all forms is to be performed. When trying to find the answers to these questions one has to keep in mind that the graphical units composing a MWU are themselves simple words that have their own inflectional behavior. However, simple words that represent components of a MWU do not inflect freely - they have to conform to some combining rules. In the case of Serbian, these rules can be rather complex, since simple words constituting a MWU, like adjectives and nouns, can inflect in several grammatical categories.

For instance, *civilni vojni rok* 'civil military service' is a multi-word noun that inherits its gender from the constituent noun *rok* (masculine in this case), and it inflects for case, but it does not inflect for number (although the simple word *rok* does). The adjectives *civilni* and *vojni* agree with the noun *rok* in number, case, gender and animacy, but the comparative and indefinite forms of these adjectives are not used in this MWU. It is clear from this example that the combining rules for Serbian MWUs are by no means trivial. After considering several options, we concluded that Multiflex, a finite-state tool for MWUs, developed by A. Savary [5], suits our needs best. This tool supports finite-state transducers that can model a number of combining conditions. As for the inflection of constituent simple words, it relies completely on FSTs for the inflection of simple words. This way, the generation of all inflected forms with Multiflex is performed with two types of FSTs: for inflection of simple words and for inflection of MWUs. To that end, a lemma for each simple word constituent that inflects in a MWU has to be provided, as well as values of all grammatical categories of forms appearing in the MWU lemma (its regular or dictionary form). For the given example, the entry in DELAC dictionary is:

civilni(civilni.A2:adms1g) vojni(vojni.A2:adms1g) rok(rok.N81u:ms1q),NC_AXAXN1

The information given in this entry allows automatic production of all 26 inflected forms for the DELACF dictionary as, for example, the entry for the singular dative case:

civilnom vojnom roku,civilni vojni rok.N:ms3q

Although the Multiflex approach is theoretically well-founded, easy to understand and apply, and it successfully solves many problems of MWU inflection for Serbian, it is obvious from the given example that the production of a single DELAC entry is a very tedious task. As a matter of fact, we initially produced only 30 DELAC entries from scratch. Realizing that additional information within DELAC entries (everything between the parenthesis) in most cases already exists in dictionaries of simple words (DELA) we decided to develop a module for our lexical resources management tool LeXimir, an enhancement of its predecessor WS4LR [6], that would help in obtaining this information. However, due to homography of forms and homonymy of simple word entries, the developer of a DELAC entry still needs to choose between several options offered by a dictionary look-up provided by this tool. For the above example, the choice had to be made three times for the information about forms (like 'adms1g' for the first component) and once for the simple word entry (because there are two entries for *rok* in the Serbian DELAS: one for 'service' and one for 'rock music'). Following this approach, only 3195 DELAC entries were produced in the past three years, which we found very ineffective.

In a comprehensive analysis of several tools for MWU inflection description [3] the author mentions only one system, *FASTR*, that supports automated MWU lexicon creation [7]. Since this system is based on an approach very different from DELA methodology, we developed our own procedure for automatic construction of DELAC entries from a given list of MWUs. For an item such as *civilni vojni rok* this procedure produces the aforementioned MWU lemma. However, the produced list of lemmas has to be manually checked and some decisions still have to be made even when the procedure offers only one candidate MWU lemma. Thus we have to stress that our dictionaries remain handcrafted resources (as per categorization in [8]) in terms of the information they offer and the methodology used to produce them (without statistical engineering).

2 Analysis of Initial Data

We based the development of the automated procedure for DELAC construction on the initial dictionary of MWUs, which contained 3195 lemmas covering different part of speech as presented in Table 1: nouns, adjectives, adverbs, conjunctions, prepositions, interjections. As only MWU nouns and adjectives inflect, they have an inflectional class code assigned to them in DELAC. Each inflectional class is associated with one inflectional transducer (as described in [9]) that controls the production of all inflected forms. The forms/lemmas ratio in Table 1 shows that a direct production of DELACF dictionary is out of the question for Serbian, though it may seem possible for some other languages. For instance, for English this ratio for MWU nouns is 1.90, whereas for French it is 1.29. The calculation is performed on the basis of dictionaries described in [10] and [11] that are part of the standard distribution of Unitex [12], a corpus processing system based on the finite-state technology.

Some inflectional transducers are frequently needed (like NC_AXN, which is used for MWUs consisting of an adjective followed by a noun, in our case 1208

Table 1. Initial content of the Serbian morphological dictionary of MWUs

POS	lemmas	forms	forms/lemmas	Inflectional classes	Inflectional super-classes
Nouns	2571	49792	19.4	67	28
Adjectives	207	21045	101.7	16	9
Other	415				

MWUs), while some apply to a single MWU. In order to design our automated procedure we grouped all inflectional transducers into equivalence classes or super-classes: a super-class consists of all transducers that use the same form of a MWU lemma, that is, the same information for the production of inflectional forms. This is also reflected in the convention we used for naming the inflectional transducers: A stands for an adjective constituent, N stands for a noun constituent, X stands for a constituent that does not inflect (including a separator), while some additional digits and letters may be added to differentiate transducers. This is illustrated in Table 2 by six inflectional transducers all belonging to one super-class NXN and used for the inflection of MWUs consisting of a noun followed by another noun, where both nouns inflect and must agree in basic grammatical categories.

It should be noted that MWUs sharing the same inflectional class (or super-class) do not necessarily have the same syntactic structure and vice versa. For instance, *film u boji* 'color movie' and *bolest ludih krava* 'mad cows disease' share the same class NC_N4X, although the first MWU consists of a noun followed by a prepositional phrase and the second of a noun followed by a phrase in the genitive. However, from the inflectional point of view they both behave in the same way: only the first component inflects, whereas the second and the third do not. On the other hand, *predsednik države* 'president of the state' and *advokat odbrane* 'attorney of the defense' both consist of a noun followed by a component in the genitive case, but their inflectional behavior is different: in the first MWU the second component can change in number whereas in the second MWU it does not inflect. For that reason they belong to two different classes (and super-classes), NC_NXN4 and NC_N4X respectively.

In order to formulate our strategy for the production of MWU lemmas we analyzed the data in the initial dictionary looking for useful information. We identified the additional information assigned to components of MWUs belonging to a particular inflectional class, and vice versa, we identified inflectional classes associated with the same additional information. For instance, the class NC_NXN2m explained in Table 2 is associated with only one combination of grammatical categories, characterized by the fact that the components differ in gender. On the other hand, some combinations of grammatical categories like [ms1q,ms1q] (two masculine inanimate nouns in the nominative singular) were associated in DELAC with three classes: NXN, NXN2, and NXN3 (see Table 2), the first one being the most frequent.

Table 2. The inflectional classes for MWUs belonging to the super-class NXN

Infl. class	Example	Translation	Explanation
NC_NXN	lekar akušer	obstetrician	Both components inflect and agree in case and number
NC_NXNF	kit ubica	killer whale	The gender of the second component changes in plural
NC_NXN3	kamen-temeljac	foundation stone	The separating hyphen can be omitted
NC_NXN2	Kongo Brazavil	Congo-Brazzaville	Neither of the components inflects for number
NC_NXN4	predsednik države	president of the state	The first component inflects for case and number; the second may or may not inflect for number
NC_NXN2m	Kneževina Monako	Principality of Monaco	The second component does not necessarily inflect

On the basis of this information we got the general idea which combinations of atomic data can we expect to be extracted from dictionaries of simple words and how they combine with inflectional classes. However, the obtained information was incomplete. For components that do not inflect for a certain MWU there was no additional information in the MWU lemma. For such components this information was subsequently searched in dictionaries of simple words. It was thus possible to establish that in *naliv-pero* 'ink pen' the first component was not in the dictionary of simple words, in *bruto plata* 'gross income' the first component was an adjective that does not inflect, in *mini-suknja* 'mini-skirt' the first component was a prefix. All three examples belong to the same super-class 2XN. There are still cases when no additional information, whether extracted from MWU lemmas themselves or subsequently from electronic dictionaries, can help in deciding in favor of one inflectional class over other possibilities. For instance, this is the case for *krem-karamel* 'caramel cream' and *kamen-temeljac* 'foundation stone': both components in these MWUs are masculine inanimate nouns in the nominative singular, but in the first one *karamel* is the head and the first component does not inflect (super-class 2XN), while in the second example *kamen* is the head and both components inflect (super-class NXN).

3 Description of the Strategy

The purpose of the analysis performed in the first step was not to produce a strategy for construction of MWU lemmas automatically — it was rather used as a reference during the manual production of a strategy in the form of XML documents. The schema of these documents is presented in Figure 1. Our strategy consists of two XML documents, one for MWU nouns, and the other for MWU adjectives. Each XML document consists of a sequence of rules that are grouped by the number of components in MWUs. Each rule states the conditions that a MWU and its components have to satisfy in order to be placed in a particular inflectional class and to have a particular MWU lemma assigned to them. In order to make our strategy easier to produce and maintain, conditions for some

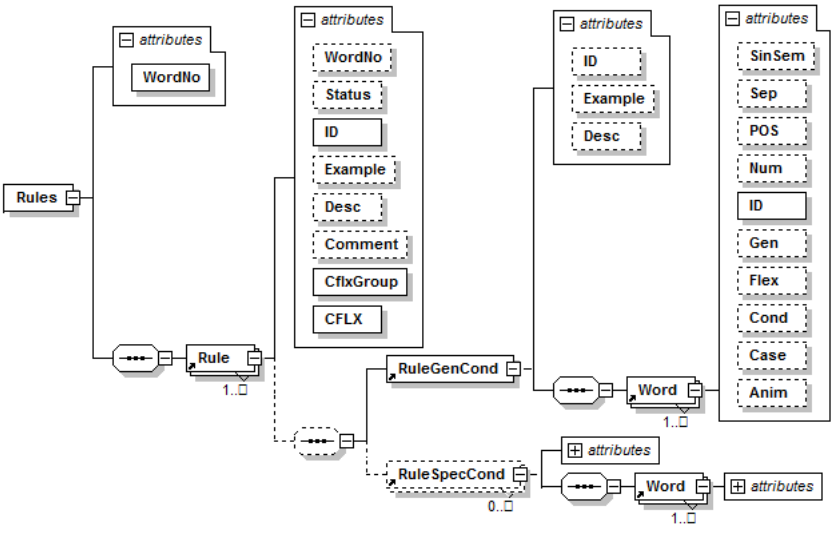


Fig. 1. The XML Schema for a strategy document

rules were grouped into general and specific conditions, where specific conditions are simply additions to general conditions. One rule will illustrate this:

```
<Rule ID="2" CFLX="NC_AXN3" CflxGroup="AXN">
 <RuleGenCond>
  <Word ID="1" POS="A" Flex="true" Case="1" Anim="$a" Gen="$g"/>
  <Word ID="2" POS="N" Flex="true" Case="1" Anim="=$a" Gen="=$g"/>
 </RuleGenCond>
 <RuleSpecCond ID="1" Example="Ajfelova kula">
  <Word ID="1" Num="s" Cond="$PRE"/>
  <Word ID="2" Num="s"/>   </RuleSpecCond>
 <RuleSpecCond ID="2" Example="poljski radovi">
  <Word ID="1" Case="1" Num="p"/> <Word ID="2" Case="1" Num="p"/>
 </RuleSpecCond>
 <RuleSpecCond ID="3" Example="elektronsko poslovanje">
  <Word ID="1" Case="1" Num="s"/>
  <Word ID="2" Case="1" Num="s" SinSem="+VN,+Coll,+HumColl"/>
 </RuleSpecCond>
</Rule>
```

This rule classifies a MWU in the inflectional class NC_AXN3 associated with adjective-noun MWUs that do not inflect for number (super-class AXN) if the following conditions are satisfied:

a) General conditions: the first component is an adjective, the second is a noun, both components are listed in the nominative case, they agree in gender and animacy (whichever they are);
b) Additional specific conditions:
   – both components are in singular and the first starts with upper-case (*Ajfelova kula* 'Eiffel tower'), or

Table 3. Overview of rules for nouns and adjectives with 2 to 7 components

Components	Rules for nouns		Rules for adjectives	
	(general)	(special)	(general)	(special)
2	26	59	5	9
3	25	85	8	8
4	14	50	5	5
5	6	54	2	2
6	4	29		
7	1	9		
Total	76	286	20	24

- both components are in plural (*mirovne snage* 'peace forces'), or
- both components are in singular and the second component is a verbal noun or a collective noun (*belo roblje* 'white slaves').

The number of rules for nouns and adjectives are listed in Table 3. Some rules have no special conditions while some have up to ten special conditions.

The rules are applied in the sequence in which they appear in the XML document, which means that if more than one rule can apply for a particular MWU, then more MWU candidate lemmas will be offered in the order of the application of rules. Hence, the order of rules in the strategy is of importance. For instance, there are three rules that can apply to the noun-noun inflectional class NC_NXN:

1. The first rule is 9^{th} in the sequence of rules and its conditions are very strict: both components have to agree in all four grammatical categories: number, case, gender and animacy (e.g. *lekar akušer* 'physician obstetrician');
2. The second rule is 21^{st} in the sequence and its conditions are less strict: the components do not have to agree in animacy (e.g. *biljka*(q) *mesožderka*(v) 'plant carnivore');
3. The third rule is 22^{nd} in the sequence and allows the components to disagree in gender (e.g. *roman*(m) *reka*(f) 'novel fleuve').

As a consequence, for noun-noun MWUs in which both nouns do not agree in all grammatical categories, the other rule producing the NC_2XN class will have precedence, as this class is, in general, more frequent.

There are some inflectional classes for which no rule can be applied: we identified 14 such classes for nouns and 2 for adjectives. This is usually related to optional use or omission of a hyphen, and in these cases the strategy offers a very similar class. There are also some classes that are related to only one very specific MWU, so we have not even tried to formulate a rule for them, e.g. the class NC_2XN3XN1 for *Sao Tome i Principe*. However, all super-classes are covered by rules.

4 Analysis and Evaluation of the Strategy

We have performed the analysis and evaluation of our strategy in two steps. In the first step we applied the strategy to the same data that we used to produce

it, that is our initial DELAC dictionary. In the second step, we applied it to several lists of MWUs that we have collected from various sources.

After applying our strategy to the initial DELAC dictionary containing 2571 nouns and 207 adjectives the obtained results were manually validated. When assessing the success of the strategy we adhered to the following:

a) If for a given MWU one of the rules produced the correct lemma and assigned the correct inflectional class, we considered the strategy as successful;
b) If for a given MWU none of the rules satisfied a), but one of the rules produced the correct lemma, although the assigned inflectional class was not correct, whereas the assigned super-class was correct, we considered the strategy as partially successful;
c) If for a given MWU none of the rules satisfied a) or b) we considered that the strategy failed for that MWU;
d) For each lemma where either a) or b) applied, we also determined the rank of the accepted solution: the higher on the list of offered solutions, the more favorable the accepted solution.

Table 4 gives percentages of successfully produced noun and adjectives entries (case a), partially successful results (case b) and failures (case c), and also indicates the rank of these results (rank 0 means no solution offered). A failure means that either no solution was offered or none of the solutions was classified as a) or b). In either case, the reason was that the strategy failed to cover a particular MWU structure (in 52 cases, or 20% of all failures) or a MWU component that inflects was not in the dictionary of simple words (in 255 cases, or 80% of all failures). The latter case occurred frequently due to MWUs representing proper names, where components are often not words in Serbian, e.g. *Dar es Salam*, or due to the fact that some words are used only in MWUs (like *domali* in *domali prst* 'next to little (ring) finger') and are thus not listed in dictionaries of simple words. The lowest rank of a successful result was 10 for nouns, and 3 for adjectives.

In the second step we applied our strategy for nouns to several different lists of MWUs. We have not applied our strategy for adjectives in this step simply because we have not collected enough new adjectives. Our list of new MWU nouns came from several different sources: the official list of MWU names of settlements in Serbia (236), MWUs extracted from a log file of a Serbian professional journal that deals with economic issues (162), from Verne's novel *Around*

Table 4. Evaluation of the strategy applied to the initial DELAC dictionary

Rank	Nouns				Adjectives			
	a)	b)	c)	Total	a)	b)	c)	Total
0			5.50%	5.50%			2.93%	2.93%
1	63.74%	12.69%	6.28%	82.71%	37.56%	15.61%		53.17%
2	6.56%	0.62%	0.08%	7.26%	36.59%	4.39%		40.98%
3	2.22%	1.21%		3.43%	2.93%			2.93%
4-10	0.90%	0.20%		1.10%				0.00%
Total	**73.42%**	**14.72%**	**11.87%**	**100.00%**	**77.07%**	**20.00%**	**2.93%**	**100.00%**

the world in 80 days (114), from the explanatory dictionary of Serbian, under the letter R (604). As the analysis in the first step indicated that MWU proper names produce in general worse results, we decided to separate these lists in two groups. After removing those already in DELAC we got a list of MWU toponyms (206) and a list of MWU common nouns (784).

Table 5 shows that in the case of common nouns, for only 3.62% items on the list (28 items) no satisfactory solution (a or b) was offered. For toponyms this percent is much higher (38.61%), accounting for 78 items on the list. In the case of toponyms all cases of failure are due to the fact that components of toponyms were not simple words in Serbian (e.g. *Feja* in *Kriva Feja*). The lowest rank of a successful result was 6 for common nouns, and 2 for toponyms.

Table 5. Evaluation of the strategy for nouns applied on a list of MWU common nouns and compound toponyms

Rank	Common nouns				Toponyms			
	a)	b)	c)	Total	a)	b)	c)	Total
0			1.68%	1.68%			24.75%	24.75%
1	80.23%	10.08%	1.94%	92.12%	48.02%	3.47%	13.86%	65.35%
2	4.39%	0.26%		4.78%	8.91%			8.91%
3	1.29%	0.13%		1.42%				0.00%
4-6				0.00%	1.00%			1.00%
Total	**85.92%**	**10.47%**	**3.62%**	**100.00%**	**57.92%**	**3.47%**	**38.61%**	**100.00%**

Row 2 in Table 6 shows that less rules were used for common nouns in the second step than for nouns in the initial DELAC in the first step. This is probably due to the fact that while building our initial DELAC and the inflectional classes we tried to find various structurally different examples. Row 3 shows that for all subsets (except adjectives) the number of rules that were not used is greater than the number of used rules, namely, many rules were added to the strategy upon analogy with other rules. As the number of rules does not affect the effectiveness of the procedure (except the processing time to a small degree) we believe that unused rules should not be removed because they may prove useful in the future. Indeed, the subset "common nouns" used seven rules that were not used for the initial DELAC nouns, while "toponyms" used one new rule.

The rules used most frequently for nouns (row 5) are rules pertaining to adjective-noun MWUs, which are also the most frequent in the dictionary. The rules that failed each time they were used (row 7) are obvious candidates for deletion from the strategy. Rules that were more unsuccessful than successful (row 8) should probably also be reconsidered, as for instance, by reinforcing the conditions, if possible.

The average number of solutions per item is rather low (row 9), ranging from 1.4 for the "common nouns" to 3.7 for "toponyms". In some cases much larger sets of solutions were offered. The leader is *Velika Plana* (a small town in Serbia) with 52 solutions offered. Such a high number of solutions occurred due to the homography of both components. However, even in this case the correct solution

Table 6. Strategy rules performance data for subsets: **N** (initial DELAC nouns), **A** (initial DELAC adjectives), **CN** (additional common nouns), **T** (additional toponyms)

	N	**A**	**CN**	**T**
1. Items	2571	207	784	206
2. Rules applied	85	19	56	33
3. Rules not applied	201	5	230	253
4. Applications of rules	4083	434	1060	769
5. Most frequently used rule	NC_AXN	AC_2X2	NC_AXN	NC_AXN3
(number of times applied)	1499	108	589	144
6. Absolutely successful rules	26	9	19	1
7. Absolutely unsuccessful rules	9	1	16	26
8. Rules more unsuccessful than successful	36	6	23	29
9. Solutions/item	1.6	2.1	1.4	3.7
10. Maximum solutions per item	38	6	19	52
11. Success 1^{st} solution	80.64%	54.77%	91.85%	68.43%
12. Success 2^{nd} solution	7.58%	42.21%	4.73%	11.84%
13. Success 3^{rd} solution	3.62%	3.02%	1.44%	0.0%

had a high rank: namely, the second solution for *Velika Plana* was correct. It may seem reasonable to exclude some dictionaries of simple words from this procedure, for instance, dictionaries of personal names, in order to alleviate similar problems. However, that might not be such a good idea: these very dictionaries successfully processed many items, among them three very specific names of small towns in Serbia named after famous Serbian poets and politicians: *Aleksa Šantić*, *Svetozar Miletić*, *Jaša Tomić*.

Successful outcomes, that is solutions classified as a) or b) offered at the first, second or third place (rows 11, 12 and 13) were counted for cases where the strategy offered at least one solution (not necessarily an acceptable one). These data show that for nouns, in general, if a solution is offered, it is very likely that the first one will be correct. For adjectives, the most likely solution will be among the first two offered.

5 Implementation and Usefulness

Our procedure for automatic production of DELAC entries is a module of the LeXimir tool [6], which is written in C# and operates on the .NET platform. It supports development, maintenance and exploitation of various resources: e-dictionaries, wordnets, and aligned texts. A user of this tool need not use all of these resources, or even possess them, but those that exist are visible in all modules and can be exploited in a useful way.

The e-dictionaries of simple and MWU words that we develop using LeXimir are used primarily within the Unitex system. As Unitex is open source software distributed under the terms of LGPL, we easily incorporated its modules in LeXimir for many tasks that involve manipulation of e-dictionaries, including dictionary look-up used in the module for (automated) production of DELAC

entries. To manipulate the strategy in the form of XML documents our tool relies on W3C standard languages XQuery and XSLT supported by .NET.

The user interface of the module for automatic production of DELAC lemmas is very friendly. A user can choose files with lists of MWUs and a strategy, and results are presented in a form of a table in which the user has only to check the correct solutions upon which a list of DELAC entries is produced. Various debugging tools and preference selections are at the user's disposal.

It has already been shown that LeXimir can be used for languages other than Serbian and English. Our new module for production of DELAC entries can also be successfully applied without any modification to other languages that have dictionaries of simple words in DELA (thus, directly supported by Unitex); naturally, corresponding XML documents representing the strategy for a particular language need to be created. However, the system can be easily modified to support other formats of simple words dictionaries because only the dictionary look-up module has to be changed. The experiment is already in progress for Polish, for which Multiflex is used for inflection of MWUs but simple word dictionaries are handled in a different, database environment [13].

Another tool WS4QE (shortened for Work Station for Query Expansion) was developed on basis of LeXimir, and it enables expansion of queries submitted to the Google search engine [6]. Integrated lexical resources enable modifications of user queries for both monolingual and multi-lingual search. The main feature of WS4QE is that it enables inflection of simple words and MWUs supplied as key-words to Google. Again, Unitex and Multiflex are used for inflection. However, WS4QE goes one step further: for free phrases supplied as key-words having a structure covered by a MWU inflectional class, the tool uses our strategy and acts upon the first offered solution, which is the correct one in most cases.

6 Conclusions

The results that we have presented justify the efforts invested in designing our procedure because it allows for massive production of DELAC entries. We have already prepared a list of 25,000 MWUs extracted from the Serbian explanatory dictionary that we hope to be able to process in a few months. One important thing that remains to be done is the addition of semantic and/or domain markers to MWU lemmas, which have so far been systematically added only for proper names following the approach suggested in [14]. We are considering several solutions, including one proposed in [15].

We envisage further development of our procedure. We would like to allow MWUs to be components of other MWUs (and components of free phrases as well). This would keep the number of possible structures low, and consequently reduce the number of inflectional classes and the number of rules in the strategy. For instance, a lemma for a MWU from the beginning of this article *civilni vojni rok*, could in this case be:

civilni(civilni.A2:adms1g) {vojni rok}(vojni rok.NC_AXN:ms1q),NC_AXN

That is, its second component could be a MWU itself (*vojni rok* 'military service') and not a simple word and it would remain in the most frequent adjective-noun class. This approach is already implemented in Multiflex but not in Unitex. However, DELAC entries that are already produced need not be revised.

References

1. Courtois, B., Silberztein, M.: Dictionnaires électroniques du français. Larousse, Paris (1990)
2. Krstev, C.: Processing of Serbian - Automata, Texts and Electronic Dictionaries. Faculty of Philology, University of Belgrade, Belgrade (2008)
3. Savary, A.: Computational Inflection of Multi-Word Units - A Contrastive Study of Lexical Approaches. Linguistic Issues in Language Technologies 1(2) (2008)
4. Krstev, C., Vitas, D.: Finite State Transducers for Recognition and Generation of Compound Words. In: Erjavec, T., Žganec Gros, J. (eds.) IS-LTC 2006, Ljubljana, Slovenia, Institut Jožef Stefan, pp. 192–197 (October 2006)
5. Savary, A.: Multiflex: A Multilingual Finite-State Tool for Multi-Word Units. In: Maneth, S. (ed.) Implementation and Application of Automata. LNCS, vol. 5642, pp. 237–240. Springer, Heidelberg (2009)
6. Krstev, C., Stanković, R., Vitas, D., Obradović, I.: The Usage of Various Lexical Resources and Tools to Improve the Performance of Web Search Engines. In: 6th LREC, Marrakech, Marocco (2008)
7. Jacquemin, C.: Spotting and Discovering Terms through Natural Language Processing. MIT Press, Cambridge (2001)
8. Laporte, E.: Lexicons and Grammars for Language Processing: Industrial or Handcrafted Products? In: Rezende, L.M., da Silva, B.C.D., Barbosa, J.B. (eds.) Léxico e gramática: dos sentidos à construção da significação. Trilhas Lingüísticas, vol. 16, pp. 51–84. Cultura Acadêmica, São Paulo (2009)
9. Krstev, C., Vitas, D., Savary, A.: Prerequisites for a Comprehensive Dictionary of Serbian Compounds. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 552–563. Springer, Heidelberg (2006)
10. Savary, A.: Recensement et description des mots composés - méthodes et applications. PhD thesis, Université de Marne-la-Vallée (2000)
11. Courtois, B., Garrigues, M., Gross, G., Gross, M., Jung, R., Mathieu-Colas, M., Silberztein, M., Vivès, R.: Dictionnaire électronique des noms composés DELAC: les composants NA et NN. Technical Report 55, LADL, Université Paris 7 (1997)
12. Paumier, S.: Unitex 2.1 User Manual (2008), http://www-igm.univ-mlv.fr/unitex/UnitexManual2.1.pdf
13. Wolinski, M., Savary, A., Sikora, P., Marciniak, M.: Usability Improvements in the Lexicographic Framework Toposlaw. In: Vetulani, Z. (ed.) 4th LTC, Poznań, Poland, IMPRESJA Widawnictwa Elektroniczne S.A (2009)
14. Grass, T., Maurel, D., Piton, O.: Description of a Multilingual Database of Proper Names. In: Ranchhod, E., Mamede, N.J. (eds.) PorTAL 2002. LNCS (LNAI), vol. 2389, pp. 137–140. Springer, Heidelberg (2002)
15. Elia, A.: The Electronic Thematic Linguistic Atlases (Atlanti Linguistici Tematici Informatici - ALTI. In: Atlas DICoMP - Dizionario delle parole composite, http://www.ricercaitaliana.it/prin/unita_op_en-2005109535_003.htm

Collocation Extraction in Turkish Texts Using Statistical Methods

Senem Kumova Metin[1] and Bahar Karaoğlan[2]

[1] Izmir University of Economics,
Engineering and Computer Science Faculty, Izmir, Turkey
senem.kumova@ieu.edu.tr
[2] Ege University, International Computing Institute, Izmir, Turkey
bahar.karaoglan@ege.edu.tr

Abstract. Collocation is the combination of words in which words appear together more often than by chance. Since collocations are blocks of meaning, they play an important role in natural language processing applications (word sense disambiguation, part of speech tagging, machine translation, etc). In this study, a corpus of Turkish is subjected to the following statistical techniques: frequency of occurrence, mutual information and hypothesis tests. We have utilized both stemmed and surface form of corpus to explore the effect of stemming in collocation extraction. The techniques are evaluated by recall and precision measures. Chi-square hypothesis test and mutual information methods have produced better results compared to other methods on Turkish corpus. In addition, we have found that a stemmed corpus facilitates discrimination between successful and unsuccessful collocation extraction methods.

Keywords: Collocation, collocation extraction.

1 Introduction

Collocations are conventional word combinations that co-occur together so recurrently that they may not be regarded as random combination of words. The term collocation has been first introduced by an English linguist, J. R. Firth, in the book"Modes of Meaning" [1] in which he states that a word can be understood by the company it keeps and gives some examples to illustrate the notion of collocations. In his further study, he states "Collocations of a given word are statements of the habitual or customary places of that word". Later, Sinclair, a student of Firth, defined collocation as the occurrence of two or more words within a short space of each other in a text [2]. In contrast, Hoey [3] gives a more statistical definition, stating that a collocation is the appearance of two or more lexical items together with a probability that cannot be interpreted as random. In Oxford Collocation Dictionary collocation is defined as the co-occurrence of words to produce natural-sounding speech and writing.

Since collocations are arbitrary and indefinite, they have an important effect on meaning in text and speech. As a result, extracting of collocations supports a

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 238–249, 2010.

wide range of natural language processing applications such as natural language generation, machine translation, word sense disambiguation, part of speech tagging, information retrieval, computational lexicography, corpus linguistic search and in some social studies through language ([4], [5]). In order to serve for this wide range of applications, many different methods of collocation exploration can be found in the literature which can be categorized as: statistical and rule based methods. Rule based methods depend especially on part of speech tagging information. On the other hand, statistical methods (frequency measure, mutual information [6], hypothesis testing, etc.) are based on some kind of frequency measure to extract collocations in a given corpus. Smadja's Xtract [7] and the techniques of Kita et al. [8] and Shimohata et al. [9] are also examples of methods known by the names of researchers.

In this study, we have applied some statistical techniques to extract collocations in Turkish and compared the results using recall and precision measures. We have utilized both stemmed and surface-formed corpora to examine the effect of extensive agglutination in Turkish. Although there are many studies on different languages, including English, Spanish, Russian, Chinese, French, to the best of our knowledge, there is no corresponding study on Turkish in this concept. We believe that our results may further open research in the field of agglutinative languages, especially Turkish.

In section 2, the term "collocation" is presented. In section 3, we have given previous work on Turkish collocations. In section 4, collocation extraction techniques which are implemented in the study are briefly described. In section 5, experimental setup which clarifies utilized corpora, the base set and evaluation method is given. Section 6 involves the implementation results. Finally section 7 deals with the discussion of the above study.

2 Collocation

As it is evident from different definitions of collocation in recent works, there are no known rules for the formation of collocations. Although researchers do not have a total consensus on either the definition of collocation or the rules by which they are created, common features collected from different studies may be listed and defined as in below:

Collocations are recurrent. Of all properties which discriminates collocations from other word combinations, recurrence is the easiest property to measure. As a result, almost all extraction techniques depend on some kind of frequency measure ([4],[6], [7], [10], [11]).

Collocations are arbitrary and language specific. There are no known rules that define which words collocate and how a word chooses a particular word or words from millions of different words in language to create a collocation. For example "strong" is a common collocation with "coffee" in English, But there is no clear explanation for the preference of this word instead of "powerful". Also collocations may change in different languages depending on the social or

cultural behaviors of native speakers. In Turkish, "strong coffee" is called "sert kahve", the exact translation of the words to English gives "hard coffee".

Collocations create a unit block in language. In natural language processing applications considering sense or meaning integrity, a unit block may be defined as a single word or a combination of words that has an individual meaning/sense and may be regarded as a sentence or a constituent of a sentence. Especially in applications such as word sense disambiguation, part of speech tagging or machine translation, detection of units is an important preprocessing step that affects the whole performance of the proposed system. For example, the collocation "lady killer" means a man exceptionally attractive to women, rather than one who kills them. So for collocations, the meaning of the whole is not the meaning of the parts.

Collocations are domain dependent. There are many different domain specific collocations especially in particular sports, medicine or science. Smadja [7] gave the domain of sailing as an example. Word combinations a "dry suit" or "a wet suit" does not mean a suit that is literally dry or wet; they are special types of suit which sailors use, but these meanings are not obvious for even native speakers.

Since the definition of collocation is still a controversial issue, in our study, we assumed the following word combinations as collocations:

- Compound verbs (e.g. take over.)
- Frequently used rigid noun phrases, domain specific terms (e.g. strong coffee, natural language processing)
- Frequently used word combinations and conjunctions (e.g. ad hoc, strong enough, as soon as, no way)
- Named entities, personal names, job titles, abbreviations (e.g. White House, general manager, Prof. Dr.)

Although collocations may involve two or more words with or without other words in between, in this study we focused on consecutive two-word collocations due to time and space complexity issues. From now on, a collocation is regarded as a group of two consecutive words (bigrams) with above listed features.

3 Collocation Extraction in Turkish

Turkish is a highly productive language through extensive agglutination, with a rich set of derivational and inflectional suffixes. In a theoretic manner, it is possible to derive millions of different word forms from just a particular lexeme in Turkish. As a result, computational linguistic applications have a high level of complexity of time and space. In addition to this high level of complexity in applications, language models may need modifications for surface formed corpus and stemmed corpus.

Recent work on Turkish collocations involves commonly linguistic studies discussing the importance of collocation notion in translation and second language education or examining the collocativity of a particular word in written Turkish texts ([12],[13],[14]).

In the area of computational linguistics, Oflazer et. al. [15] propose a rule-based multiword expression processor. The processor extracts multiword expressions in a morphologically analyzed corpus in which parts of speech and inflections are all tagged. Multiword expressions are categorized in four different forms in the study by Oflazer et. al.: lexicalized collocations, semi-lexicalized collocations, non-lexicalized collocations and multiword named entities. Depending on the certain morphological patterns, multiword collocations are retrieved by querying about 1100 rules.

4 Some Collocation Extraction Techniques

There are various statistical and rule based techniques for collocation extraction. We have applied common statistical techniques to avoid the time and space complexity of the preprocessing steps needed in rule based techniques. In the following subsections, utilized techniques; frequency of occurrence, mutual information and some of hypothesis tests: t-test, log-likelihood, chi-square; will be briefly described.

4.1 Frequency of Occurrence

The frequency of occurrence method is the simplest and earliest approach on collocation extraction. In this technique, the frequency of bigrams or frequency of words that co-occur in a given window designates whether the word combination is a collocation or not. Combinations are ranked by the frequency to create a candidate list, and the most frequent combinations are accepted as collocations. Although some of the frequent candidates are collocations, others are pairs of function words [5]. To discard these frequent function word pairs, filtering (e.g part of speech tagging) is recommended in many existing studies [16].

4.2 Mutual Information

In information theory, mutual information is defined as the quantity that measures the mutual dependence of the two variables. In collocation extraction instead of two random variables, the definition is modified for values of random variables. Thus, a new measure, point-wise mutual information, is introduced ([6],[10]). If we write x and y for the first and the second word respectively, point-wise mutual information for them is given by

$$I(x, y) = log_2 \frac{P(x, y)}{P(x) \cdot P(y)} \tag{1}$$

If the words x and y are independent of each other; the probability of the words coming together must be equal to the multiplication of their own probabilities

$(P(x,y) = P(x) \cdot P(y))$. In this case, mutual information will be zero $(I(x,y) = 0)$ indicating that words do not collocate. Therefore the further the mutual information of a combination moves away from zero, the closer it becomes to being a collocation

4.3 Hypothesis Testing

To decide whether a word combination is a collocation, it is necessary to prove that the joint occurrence of the words is more than coincidence. The common approach showing the dependence between words is testing the hypothesis of independence. Hypothesis testing methods attempt to reject the null hypothesis that states that words in combination are independent of each other. The different methods testing null hypothesis used in literature are described as follows:

4.4 Dunning's Log-Likelihood Test

Log-likelihood method is a hypothesis testing approach presented by Dunning [11]. In collocation discovery, two alternative explanations for the occurrence frequency of bigram $w1w2$ is examined:

Hypothesis 1: $P(w2/w1) = p = P(w2/\neg w1)$
Hypothesis 2: $P(w2/w1) = p1 \neq p2 = P(w2/\neg w1)$

Hypothesis 1 formulates that the occurrence of $w2$ is independent of the previous occurrence of $w1$. In contrary, hypothesis 2 formalizes dependence of word combination. If hypothesis 1 is accepted, the combination is not a collocation and if hypothesis 2 is accepted, combination is regarded as a collocation. The maximum likelihood estimates for p, p_1 and p_2 are given as below.

$$p = \frac{c}{N}$$

$$p_1 = \frac{c_{12}}{c_1}$$

$$p_2 = \frac{c_2 - c_{12}}{N - c_1}$$

c_1, c_2, c_{12} are respectively the number of occurrences of $w1$, $w2$ and $w1w2$; N is the total number of words in the corpus. Assuming a binomial distribution $(b(k;n,x) = x^k(1-x)^{n-k}$) log-likelihood ratio, λ, is then as follows :

$$log\lambda = log\frac{L(H1)}{L(H2)} = log\frac{b(c_{12},c_1,p)b(c_2-c_{12},N-c_1,p)}{b(c_{12},c_1,p_1)b(c_2-c_{12},N-c_1,p_2)} \tag{2}$$

$$\begin{aligned} &= logL(c_{12},c_1,p) + logL(c_2-c_{12},N-c_1,p) \\ &\quad -logL(c_{12},c_1,p_1) - logL(c_2-c_{12},N-c_1,p_2) \end{aligned}$$

where $L(k,n,x) = x^k(1-x)^{n-k}$. Mood (1974) has shown that $-2log\lambda$ has an asymptotically χ^2 distribution. So if the calculated values $-2log\lambda$ are less than

χ^2 value at given level of significance, the null hypothesis of independence is accepted otherwise the hypothesis that states $w1w2$ is a collocation is accepted. As a result, the log-likelihood method produces a statistic that tells how much more likely one hypothesis is than the other; the higher the number the closer the candidate is to being a collocations. The method is applied to all word combinations in the corpus and a ranked list is generated to extract collocations.

4.5 The t-Test

In the t-test, null hypothesis states that sample is drawn from a normal distribution with mean μ. The test looks at the differences between expected and observed means scaled by variance of the data. As a result, if the observed mean differs from expected mean, null hypothesis is rejected. The t statistics is computed as

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}} \tag{3}$$

where $\bar{x}$ is the sample mean (observed mean), s^2 is variance, N is the sample size and μ is the mean of the distribution.

To apply the t-test for independence of two words $w1$ and $w2$ assuming that $f(w1)$,$f(w2)$ and $f(w1w2)$ are respective frequencies of $w1$,$w2$ and $w1w2$ and N is the total number of words in the corpus, following probabilities may be given

$$P(w1) = \frac{f(w1)}{N}$$

$$P(w2) = \frac{f(w2)}{N}$$

$$P(w1w2) = \frac{f(w1w2)}{N}$$

The null hypothesis is $P(w1w2) = P(w1) \cdot P(w2)$ in the test and if the null hypothesis is true then randomly generating bigrams of words and assigning 1 to the outcome $w1w2$ and 0 to any other outcome follows a Bernoulli distribution with mean $\mu = P(w1) \cdot P(w2)$. Using binomial distribution sample mean is $\bar{x} = P(w1w2)$ and sample variance is $s^2 = P(w1w2)(1 - P(w1w2)) \approx P(w1w2)$ since $P(w1w2)$ is small for most word combinations. The t-values for all word combinations are calculated and compared with the t-value at a predefined level of significance. Null hypothesis is rejected for the combinations that have higher t-values than the values in t-table.

4.6 Chi-Square (χ^2) Test

χ^2 test is a hypothesis testing technique presented by Pearson. The technique does not require normally distributed probabilities as in t-test. The test is applied to 2x2 tables to compare observed frequencies with expected frequencies

Table 1. 2x2 table showing frequencies for words "beyaz" and "saray"

	$w1 = beyaz$	$w1 \neq beyaz$
$w2 = saray$	8 (beyaz saray)	4667 (e.g, kervan saray)
$w2 \neq saray$	15820 (e.g, beyaz tül)	1428781 (e.g, kedi tüyü)

to examine whether the null hypothesis of independence can be rejected. The expected frequency representing independence are calculated and if the observed frequencies differ from expected frequency; null hypothesis is rejected. Table 1 includes the 2x2 frequency table of the words "beyaz" (white), "saray" (palace). The couple beyaz "saray" refers to "The White House" in English. The null hypothesis in χ^2 test may be stated as "beyaz" and "saray" are independent.

The χ^2 statistic sums differences between observed (O_{ij}) and expected values (E_{ij}) in all cells of the table and scales the differences by the magnitude of the expected, as follows:

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4}$$

where i is row and j is column index in the table. The expected frequency of each cell is computed from the totals of rows and columns converted into proportions.

The χ^2 value is calculated for all word combinations in corpus and a ranked list is generated. The combinations having higher values are accepted as being collocations.

5 Experimental Setup

5.1 The Base Data Set

Collocation extraction is a corpus based application. As a result, ideally a collocation tagged corpus is required. Due to the great sizes of corpora, however it is impossible to tag manually all collocations in a corpus. In many studies researchers extract a base set from corpora and implement the methods on this set. The base set may be constructed in many different ways. It may be constructed from a specific word combination considering part of speeches. For example, adjective-noun pairs in the corpora may be selected to create a base set as in the study of Evert and Krenn [17]. Or the set may be retrieved from a dictionary [18].

In our study, we have used a different approach to generate the base set. We have retrieved all bigrams in the corpus excluding those across sentence boundaries. Afterwards, five techniques defined in previous chapter are applied to generate a ranked list of bigrams. In the ranked lists, candidates having higher scores are assumed to be collocations. We selected the first (best) 200 candidates in each list to create the base set and tagged the set manually. Thus, it not only became

possible to compare all methods based on the same data set, but also all preprocessing steps to retrieve collocation candidates in the corpus were eliminated.

This study utilized the Bilkent corpus, compiled as Bilkent University for computational linguistic studies [19]. The corpus consists of articles from popular newspapers over an interval of several years. It has been morphologically analyzed by a finite state machine; sentence boundaries and stemmed forms of words have been tagged automatically [19]. We have applied collocation extraction techniques to both the stemmed and surface form of corpus to examine the effect of stemming in Turkish. The corpus has about 719665 words and 48268 sentences.

Since collocations are defined as frequently occurring word combinations, we have eliminated word combinations occurring less than four times in the corpus before the application of statistical extraction techniques.

5.2 Evaluation Method

Evaluation of extraction techniques defined in previous chapters is performed by recall and precision which are frequently used as performance measures in information retrieval. For collocation extraction, recall may be defined as the fraction of the collocations in the corpus or the base set that is successfully retrieved. Precision is the fraction of true collocations in the retrieved list of collocation candidates. Taking ϵ to be collocations extracted from base set, and δ to be the number of true collocations in the base set, recall and precision may be defined as

$$r = \frac{|\epsilon \cap \delta|}{|\delta|}$$

$$p = \frac{|\epsilon \cap \delta|}{|\epsilon|}$$

While presenting the precision and recall values, we have applied the approach of Evert and Krenn [17]. In the approach, instead of computing the measures for only a single proportion of candidate list (for example just for the whole set or just for the first N candidates), recall and precision are computed for N highest ranked candidates where N may vary from 1 to the total number of candidates ($N = 1, 2, 3, \ldots$,base data set). This approach prevents misleading conclusions being drawn from a single value of N. We have plotted graphs of precision and recall for whole base set.

6 Results

Through agglutination, in Turkish, a stem can occur in many different forms due to many possible different inflections All words in collocations are prone to agglutination, especially those in final position. As a result, in the corpus, the same collocation may occur with different surface forms and this variation reduces the total frequency of the collocation to the extent that collocation may completely lose recurrence property. For example, “maliye bakanlığı” is a collocation that

can be translated as "ministry of commerce" to English. However, this collocation may occur in the forms such as "maliye bakanlığına" (to the ministry of commerce) or "maliye bakanlığının" (of the ministry of commerce), "maliye bakanlığında" (in the ministry of commerce), "maliye bakanlığıyla" (with the ministry of commerce). Therefore the frequency of occurrence is widely spread across different forms. This property induced us to expect that collocation extraction may give better results for stemmed corpora in which different word forms are all merged to one stem.

The implementation of extraction techniques on the Bilkent corpus has returned a larger base set for stemmed corpus (661 bigrams) compared to surface formed corpus (507 bigrams), as expected. Base sets involve 53.5% and 49.8% true collocations respectively for surface and stemmed forms of the Bilkent corpus. The proportions give us the baseline for the precision graphics. As a result, if one particular method gives lower values than the baseline for a particular N value, it is said to be even worse than random selection.

Figures 1 and 2 show precision graphs of surface and stemmed Bilkent corpus, respectively. The horizontal axis in the graphs presents the percentage of base set completed. In the graphs, three important results are pointed out. Firstly, it is noteworthy that χ^2 and mutual information methods give consistently higher precision for both stem and surface form lists. In contrast, log-likelihood, t-test and frequency measure methods perform even worse than the assumed baseline which is random selection. Secondly, in the stemmed corpus, as expected, precision values are higher and the gap between χ^2 and mutual information methods with the others are more apparent.

Finally, it can be seen from the figures that the stemmed corpus gives a clearer indication of which methods are successful and which are not, and the degree of difference between them. Figures 3 and 4 show that χ^2 and mutual information methods reach higher scores of recall earlier compared to other methods supporting precision results. Correlatively, χ^2 and mutual information methods generate higher scores for true collocations and extract them earlier than other methods.

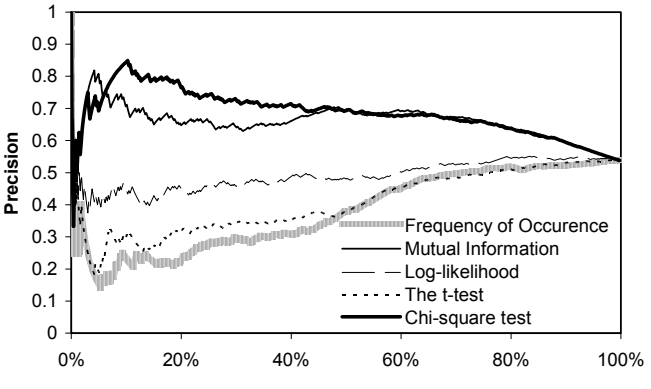


Fig. 1. Precision graph for Bilkent corpus (surface form)

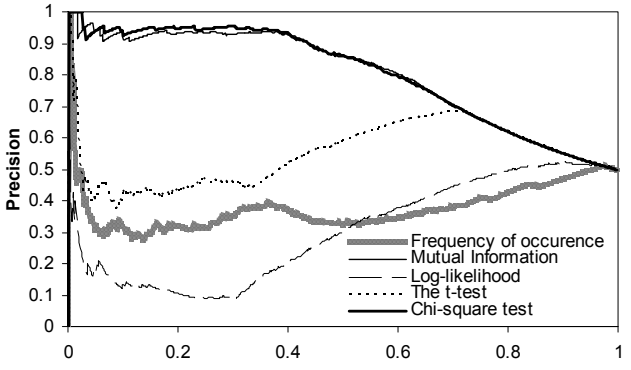


Fig. 2. Precision graph for Bilkent corpus (stemmed form)

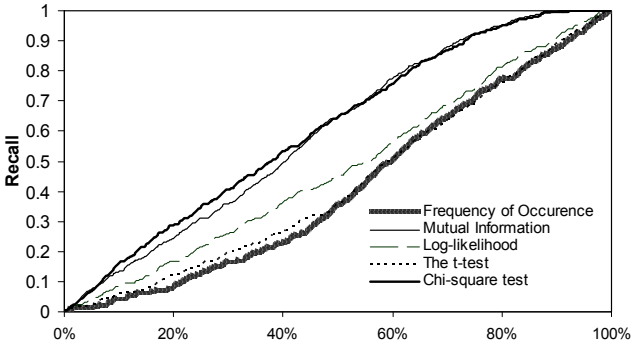


Fig. 3. Recall graph for Bilkent corpus (surface form)

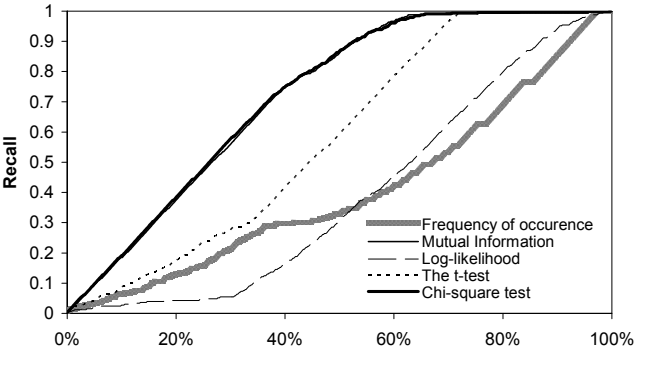


Fig. 4. Recall graph for Bilkent corpus (stemmed form)

7 Discussion

In this study some statistical methods were applied to Turkish corpus to retrieve collocations. χ^2 and mutual information methods generated higher precision and recall values compare to other techniques. Methods are both utilized on stemmed and surface form of corpus to explore the effect of agglutination in collocation extraction. It is seen that the stemmed corpus generated results that are more effective in discriminating between successful and unsuccessful methods. In a further work, we hope to be able to improve methods of analysis in the light of the results of this study.

References

1. Firth, J.R.: Modes of Meaning. Papers in Linguistics 1934-51. Oxford University Press, Oxford (1957)
2. Sinclair, J.M.: Corpus, Concordance, Collocation. Oxford University Press, Oxford (1991)
3. Hoey, M.: Patterns of Lexis in Text. Oxford University Press, Oxford (1991)
4. Bisht, R.K., Dhami, H.S., Neeraj Tiwari, N.: An evaluation of different statistical techniques of collocation extraction using a probability measure to word combinations. Journal of Quantitative Linguistics 13, 161–175 (2006)
5. Manning, C.D., Schtze, H.: Foundations of Statistical Natural Language Processing. MIT Press, England (1999)
6. Church, K.W., Hanks, P.: Word Association Norms, Mutual Information, and Lexicography. Computational Linguistics 16(1), 22–29 (1990)
7. Smadja, F.A.: Retrieving Collocations from Text: Xtract. Computational Linguistics 19(1), 143–177 (1993)
8. Kita, K., Kato, K., Omoto, T., Yano, Y.: A comparative study of automatic extraction of collocations from corpora: Mutual information vs. cost criteria. Journal of Natural Language Processing 1, 21–33 (1994)
9. Shimohata, S., Sugio, T., Nagata, J.: Retrieving collocations by co-occurrences and word order constraints. In: The Eighth Conference on European Chapter of the Association for Computational Linguistics, Madrid, Spain, pp. 476–481 (1997)
10. Hindle, D.: Noun Classification from Predicate-Argument Structures. In: Annual Meeting of the Association for Computational Linguistics (ACL 1990), Pittsburgh, Pennsylvania, USA, pp. 268–275 (1990)
11. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1), 61–74 (1993)
12. Sarıkaş, F.: Problems in Translating Collocations. Elektronik Sosyal Bilimler Dergisi 5(17), 33–40 (2006)
13. Taşıgüzel, S.: Ilköğretim T'ükçe Ders Kitaplarında Oğretici Nitelikli Metinlerdeki Eşdizimsel Orüntülerin Görünümü. Dil Dergisi. Ankara Üniversitesi Türkçe ve Yabancı Dil Araştırma ve Uygulama Merkezi 25, 72–87 (1988)
14. Özkan, B.: Türkiye Türkçesinde Belirteçlerin Fiillerle Birliktelik Kullanımlar ve Eşdizimliliği. Phd Thesis. Çukurova University. Adana, Turkey (2007)
15. Oflazer, K., Çetinoğlu, O., Say, B.: Integrating Morphology with Multi-word Expression Processing in Turkish. In: 2nd ACL Workshop on Multiword Expressions: Integrating Processing (MWE 2004), Barcelona, Spain, pp. 64–71 (2004)

16. Justeson, J.S., Katz, S.M.: Principled Disambiguation: Discriminating Adjective Senses with Modified Nouns. Computational Linguistics 21(1) (1995)
17. Evert, S., Krenn, B.: Methods for the qualitative evaluation of lexical association measures. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, Toulouse, France, pp. 188–195 (2001)
18. Pearce, D.: A comparative evaluation of collocation extraction techniques. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Spain (2002)
19. Tür, G., Hakkani-Tür, D., Oflazer, K.: A Statistical Information Extraction System for Turkish. Natural Language Engineering 9(2), 181–210 (2003)

Towards the Design and Evaluation of ROILA: A Speech Recognition Friendly Artificial Language

Omar Mubin, Christoph Bartneck, and Loe Feijs

Department of Industrial Design
Eindhoven University of Technology
Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
{o.mubin,c.bartneck,l.m.g.feijs}@tue.nl

Abstract. In our research we argue for the benefits that an artificially designed language that we call ROILA could provide to improve the accuracy of speech recognition given that it is constructed on speech recognition friendly principles. We also contemplate the trade off effect of users investing some effort in learning such a language. Initially we present the design and evaluation of the vocabulary of ROILA and subsequently we describe the ROILA grammar and the method by which we rationally chose grammar rules. Our evaluation results indicated that the vocabulary of ROILA significantly outperformed English whereas we could not yet replicate similar trends while evaluating the grammar.

Keywords: Artificial Languages, Automatic Speech Recognition, Sphinx-4.

1 Introduction

Recent research in speech recognition is gradually progressing towards altering the medium of communication in a bid to improve the quality of speech interaction. As stated in [12], constraining language is a plausible method of improving recognition accuracy. In [15] the user experience of an artificially constrained language was evaluated within a movie-information dialog interface and it was concluded that 74% of the users found the constrained language interface to be more satisfactory than natural language interface. The limitations prevailing in current automatic speech recognition technology for natural languages is an obstacle behind the unanimous acceptance of Speech Interaction. Generally in speech interfaces the focus is on using natural language; it may be time to explore a different balance in the form of a new language. The field of handwriting recognition has followed a similar road map. The first recognition systems for handheld devices, such as Apple's Newton were nearly unusable. Palm solved the problem by inventing a simplified alphabet called Graffiti which was easy to learn for users and easy to recognize for the device. Using the same analogy we aim to design a "Speech Recognition Friendly Artificial Language" (ROILA) where an artificial language as defined by the Oxford Encyclopedia is a language deliberately invented or constructed. In linguistics, there are numerous artificial

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 250–256, 2010.

languages (for e.g. Esperanto, Interlingua) whose goal is easier communication amongst users; however there has been little or no attempt to optimize a spoken artificial language for speech recognition. In summary, our research is constructed on the basis of two main goals. Firstly the artificial language should be optimized for efficient automatic speech recognition and secondly, there should be an attempt to make it learnable for a user, two possibly contradictory requirements, for e.g. users would prefer shorter words but shorter words would be harder to recognize.

2 Vocabulary Design

In order to obtain a group of phonemes that could be used to generate the vocabulary of ROILA we conducted a phonological overview of natural languages [10]. Extending from our goal of designing a language that is easy to learn for humans, we extracted a set of the most common phonemes present in the major 13 natural languages of the world based on number of speakers. We used the UCLA Phonological Segment Inventory Database (UPSID) [11]. The database provides an inventory of the phonemes of 451 languages of the world. We generated a list of phonemes that are found in 5 or more, major languages. This resulted in a total of 23 phonemes. Certain other constraints were employed to reduce this list further; diphthongs were excluded; and phonemes that had ambiguous behavior across languages were ignored. Therefore the final set of 16 phonemes that we wished to use for our artificial language was: (in ArpaBet notation) {AE, B, EH, F, IH, JH, K, L, M, N, AA, P, S, T, AH, W}.

As a starting point for the first version of the vocabulary of ROILA we choose the artificial language Toki Pona [6] which caters for the expression of very simple concepts by just 115 words. Therefore this number formed the size of the ROILA vocabulary. In order to maintain a balance between our two research goals we set the word length to 4, 5 and 6 characters, with each word having 2 or 3 syllables rendering the following word types: CVCV, VCVC, VCCV, CVCVC, VCVCV, CVCVCV, VCCVCV, VCVCCV, where V refers to a vowel and C to a consonant from our pool of 16 phonemes. The 8 word types were simple extensions of words existing in Toki Pona based on the assumption that such words would be easy to learn and pronounce. To define the scalable representation of the words we utilized a genetic algorithm that would converge to a vocabulary of words that would have the lowest confusion amongst them and in theory be ideal for speech recognition. The genetic algorithm randomly initialized a vocabulary of N words, for P vocabularies, where each word was any one of the 8 afore-mentioned word types. The algorithm was then run for G generations with mutation and cross-over being the two primary offspring generating techniques. The fitness function was determined from data available in the form of a confusion matrix (from [7]), where the matrix provided the conditional probability of recognizing a phoneme p_j by a speech recognizer when phoneme p_i was said instead. Therefore, the confusion between any two words was determined by computing the probabilistic string edit distance, as suggested

in [1]. The first ROILA vocabulary was generated by running the algorithm for P=G=200. In order to have a benchmark of English words to compare against we set the English vocabulary as the meanings of the 115 Toki Pona words.

2.1 Vocabulary Evaluation

In order to evaluate ROILA 16 (6 female) voluntary participants were asked to record samples of every word from both English and ROILA. The recordings were then passed offline through the Sphinx-4 [4] speech recognizer. Participants had various native languages but all were graduate students and hence had reasonable command over English. Recordings were carried out using a high quality microphone. Sphinx was tuned such that it was able to recognize ROILA by means of a phonetic dictionary; however an acoustic model for English was used. In addition, we did not carry out any training on the acoustic model for ROILA. One of the researchers conducted rounds of ROILA recordings until we had a pool of recordings that rendered a recognition accuracy of 100%. These sample recordings of every word would be played out before participants recorded each ROILA word. This was done to ensure that the native language of participants would not affect their ROILA articulations. The experiment was carried out as a 2 condition within subject design, where the language type (English, ROILA) was the main independent variable. The dependent variable was the total number of errors in recognition. Words from both English and ROILA were randomly presented and the order of recording English or ROILA first was also controlled between participants. We carried out a repeated measure ANOVA which revealed that language type did not have an effect $F(1,9) = 0.758$, $p = 0.41$. Both ROILA and English performed equally in terms of accuracy (67.61% and 67.66% respectively). Without any training data, such accuracy is expected from Sphinx on test data [14]. To judge if ROILA word structure had an effect on recognition accuracy, we executed an analysis in which the type of word was the independent variable. This factor had 2 levels namely CV or non-CV type, where CV-type words were CVCV, CVCVC and CVCVCV. The ANOVA analysis revealed a nearly significant trend $F(1, 113) = 3.6$, $p = 0.06$. CV-type words performed better on recognition (on average 4.19 participants got such words wrong, as compared to non CV type words, where 5.75 participants got them wrong). Therefore for our second iteration of the evaluation we generated a new vocabulary that comprised of CV type words only. The genetic algorithm was run with the parameters G=P=200. We had 11 (4 female) from the earlier 16 participants carry out recordings of the new vocabulary using the same setup and procedure. We did not have them record the English words again. Participants would once again hear sample pronunciations. The REMANOVA revealed that the new ROILA vocabulary significantly outperformed English $F(1, 10) = 4.86$, $p = 0.05$ (see Figure 1). The accuracy for the 11 participants was English: 65.11%, and ROILA_CV: 71.11%. This vocabulary was hence declared as the first ROILA vocabulary.

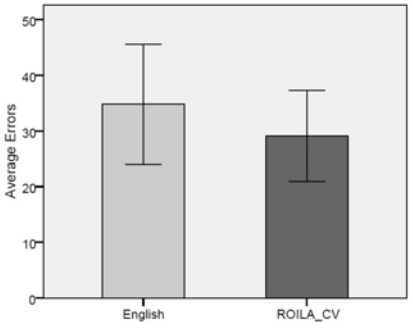


Fig. 1. Average Errors Bar Chart for English and ROILA_CV

3 Grammar Design

In conjunction with conducting a phonological overview of artificial languages we also carried out a morphological overview of artificial languages individually and also in contrast to major natural languages of the world [10]. This aided us in identifying grammar features which were popular in both sets of languages. We determined several grammatical categories based on properties defined in various linguistic encyclopedias [5]. Gender, numbering, tense and aspect are some examples. However within each category there were a number of options that we could choose from, for e.g. should we have gender? How many tenses should we have? In order to make our choice we carried out a rationale decision making process by utilizing the Questions, Options and Criteria (QOC) technique [8]. For this purpose we defined the following important criteria for every grammatical property: Learnability; defines whether the grammatical marking in question would be easy to learn or not, Expected recognition accuracy; defines the effect the grammatical marking would have on the anticipated word error rate given that the more constrained a grammar (lower perplexity) is the better it would be for recognition [9], Vocabulary size; describes the effect the grammatical marking would have on increasing or decreasing the vocabulary size, Expressive Ability of the language; defines whether using the grammatical marking in question would actually enable speakers to express more concepts then they would have been unable to do so otherwise, Efficiency; simply relates the grammatical marking to how many words would be required to communicate any solitary meaning, Acknowledgement within Natural and Artificial Languages; states the popularity of the particular grammatical marking amongst each type of languages. Appropriate weights were assigned to the criteria based on importance, for e.g. learnability and expected recognition accuracy were assigned higher weights with recognition accuracy being given twice as much weight as learnability. The total sum of the weights was 1.

Table 1. ROILA Example Sentences

ROILA Sentence	English Translation	Literal Meaning
pito fosit bubas.	I am walking to the house.	I walk house
pito fosit jifi bubas.	I walked to the house.	I walk < past tense marker > house
pito fosit jifo bubas.	I will walk to the house.	I walk < future tense marker > house

All possibilities of each grammatical category were listed and every category was then ranked across the criteria by giving a number between 1 and 3 with 3 being the best fit. The category which yielded the highest output was then chosen to be as the grammar category of choice. After filling in a matrix we concluded firstly that the ROILA grammar would be of isolating type. Affixes would not be added as this might alter the word structure hereby reducing their efficiency for speech recognition. Therefore grammatical categories in ROILA would be represented by word markers (see Table 1). At the end we arrived at the following properties: Gender (male, female) on the level of pronouns only and not nouns, Numbering (singular, plural) on the level of nouns, Person references (first, second, third) on the level of pronouns, Tense (past, present, future) and word order would be SVO.

3.1 Grammar Evaluation

In order to evaluate the grammar in terms of recognition we formulated some sample sentences (N=30) based on a hypothetical interaction scenario for a dialog system. These sentences were evaluated against their English semantic counterpart. Sphinx-4 Language Models were created using the Sphinx Knowledge Base tool [13]. An identical setup was followed as done in the evaluation of the vocabulary except that participants would now record sentences and not isolated words. Participants would once again hear a sample voice as a guide of how to pronounce sentences. The dependent variable was word accuracy, a common metric to evaluate continuous speech recognition [3] with the independent variable yet again language type. In the initial evaluation we conducted recording sessions with 8 participants. However we were unable to achieve significant results in favor of ROILA; as indicated by the REMANOVA results $F\ (1, 7) = 1.97$, $p = 0.21$.

4 Discussion and Conclusion

Our results revealed some interesting insights. Firstly, we were able to achieve improved speech recognition accuracy as compared to English for a relatively larger vocabulary. Similar endeavors have only been carried out for a vocabulary size of 10 [2]. Secondly, we quantitatively illustrated that CV type words perform better in recognition; co-articulation of CV syllables could be one explanation for that. We must keep in mind several implications to our results. Firstly, participants recorded words without any training in ROILA, whereas they were already

acquainted with English. Potentially, by training participants in ROILA the accuracy could be further improved. This effect was observed to be more pronounced when participants had to speak ROILA sentences, which could explain the insignificant difference between ROILA and English in terms of the accuracy of speech recognition. The acoustic models of Sphinx are trained with dictation training data and from what we observed the ROILA sentence articulations of participants did not fall within the domain of dictation speech. There were pauses between words and pronunciations were not smooth, which could have been caused by the inexperience of the participants in ROILA. In the future, we aim to conduct more evaluation sessions of ROILA sentences after carrying out training with additional participants. It may also be observed that the acoustic model of Sphinx was primarily designed for English, yet our ROILA accuracy in the second vocabulary iteration were significantly better as compared to English; a promising result indeed. What we would also like to determine in the future is the magnitude of the difference between ROILA and English. This could be accomplished by using the same English acoustic model for another natural language and comparing the differences in recognition accuracy between the three languages (English, ROILA and the second natural language).

We acknowledge the trade-off factor of humans having to invest some energy in learning a new language like ROILA, even though in various steps of the design process we have tried to accommodate the aspect of human learnability and introduce language features which were conducive to learnability. In summary, by designing an artificial language we are faced with the effort a user has to put in learning the language. Nevertheless, we wish to explore the benefits that an artificial language could provide if its designed such that it is speech recognition friendly. This factor might end up outweighing the price a user has to pay in learning the language and would ultimately motivate and encourage them to learn it. Another criticism that might be levied on ROILA is that many artificial languages were created already but not many people ended up speaking them. Where our approach is different is that we aim to deploy and implement our artificial language in machines and once certain machines can speak the new language it could encourage humans to speak it as well. In the future, we aim to train participants in ROILA and evaluate it by deploying it in an interaction context. We acknowledge that a meaningful societal application of our language would provide an extra gain in addition to recognition performance. We aim to explore applications for children, medical tasks, or care robots[1].

Acknowledgements. We would like to thank the reviewers for their helpful comments and feedback to revise the paper.

References

1. Amir, A., Efrat, A., Srinivasan, S.: Advances in phonetic word spotting. In: The Tenth International Conference on Information and Knowledge Management, pp. 580–582. ACM Press, New York (2001)

[1] To know more about ROILA and its latest developments please visit: http://roila.org

2. Arsoy, E., Arslan, L.: A universal human machine speech interaction language for robust speech recognition applications. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 261–267. Springer, Heidelberg (2004)
3. Boros, M., Eckert, W., Gallwitz, F., Grz, G., Hanrieder, G., Niemann, H.: Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. In: CORR (1996)
4. Carnegie-Mellon-University: Sphinx-4 (2008), http://cmusphinx.sourceforge.net/sphinx4/
5. David, C.: The cambridge encyclopedia of language (1997)
6. Kisa, S.E.: Toki pona - the language of good (2008), http://www.tokipona.org/
7. Lovitt, A., Pinto, J., Hermansky, H.: On confusions in a phoneme recognizer. IDIAP Research Report, IDIAP-RR-07-10 (2007)
8. MacLean, A., Young, R., Bellotti, V., Moran, T.: Questions, options, and criteria: Elements of design space analysis. Human-Computer Interaction 6(3), 201–250 (1991)
9. Makhoul, J., Schwartz, R.: State of the art in continuous speech recognition. Proceedings of the National Academy of Sciences 92(22), 9956–9963 (1995)
10. Mubin, O., Bartneck, C., Feijs, L.: Designing an artificial robotic interaction language. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009. LNCS, vol. 5727, pp. 851–854. Springer, Heidelberg (2009)
11. Reetz, H.: Upsid-info (2008), http://web.phonetik.uni-frankfurt.de/upsid_info.html
12. Rosenfeld, R., Olsen, D., Rudnicky, A.: Universal speech interfaces. Interactions 8(6), 34–44 (2001)
13. Rudnicky, A.: Sphinx knowledge base tool (2010), http://www.speech.cs.cmu.edu/tools/lmtool-new.html
14. Samudravijaya, K., Barot, M.: A comparison of public-domain software tools for speech recognition. In: Workshop on Spoken Language Processing, pp. 125–131. ISCA (2003)
15. Tomko, S., Rosenfeld, R.: Speech graffiti vs. natural language: Assessing the user experience. In: Proceedings of HLT/NAACL (2004)

Time Expressions Ontology for Information Seeking Dialogues in the Public Transport Domain

Agnieszka Mykowiecka

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

Abstract. The paper presents an ontology of natural language temporal expressions which occur in dialogues led by users of a public transport call center. It was elaborated on the basis of analysis of 500 transliterated dialogues and contains a multihierarchy of concepts representing semantics of time referring fragments of interlocutors' utterances. The paper contains also an analysis of frequencies of all types of time relevant expressions within the analyzed data.

Keywords: temporal expressions in Polish, time ontology.

1 Introduction

Time expressions occur in dialogues of nearly every type and for many of them resolving time references is crucial for successful communication. One of the domains for which temporal relations are extremely important is information about public transport connections. In this domain, a description of time, together with space and transportation lines organization issues, are the most important areas of interest.

As representing time points or intervals and inferring about their interdependencies is crucial for natural language understanding, no wonder that there are a lot of time models and temporal logics describing ways of operating on time points or intervals – the contemporary history of temporal logic begins from Prior (1957). To use logic inference rules, a method for representation of time related data and a way of recognizing them within a natural language text have to be established. The latter problem is quite complicated, because in natural language utterances time expressions are very frequently imprecise (e.g. 'before eight' , 'in the morning') or given in an indirect way (e.g. 'later', 'after his birthday'). There were already quite a lot of trials aimed at a definition models of time descriptions used in various contexts. There were defined a lot of time ontologies, there is also a standard of annotation of temporal expression in text – TIMEX [3]. These standards are used to annotate language corpora, e.g TimeBank, which were in turn used in temporal expressions recognition tasks (e.g. TempEval [11], [1]). TIMEX standard describes such time related notions like dates and time of

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 257–262, 2010.

a day and also imprecise notions like 'later'. However, the annotation guidelines concern English 'lexical triggers', and their usage for recognizing boarders and types of temporal phrases in texts in another natural language is limited. One of the first complex solutions of the problem of linguistic temporal expressions interpretation was proposed within the Verbmobil project for the task of appointment scheduling [2]. In recent years a lot of different ideas were explored in all the domains enumerated above [4].

The research presented here aimed at collecting and organizing linguistic material concerning temporal linguistic constructions in Polish. The motivation of this work was the fact that although in general time expressions in Polish are similar to English, their exact formulations have to be described on the basis of real data. The constructed set of concepts was used as a starting point for creating machine learning models for solving a problem of automatic recognition of temporal concepts in the chosen type of Polish dialogues. In an experiment described in [7] of semantical labeling using CRF model, for 15 types of time related concepts occurring in the test set, the F-measure ranged from 0.5 for TIME_RATE_REL to 0.97 for AT_HOUR and 1.0 for BEFORE_HOUR (only concept names without values were evaluated).

The data which is our source of knowledge about the way people talk about time in the context of using public transport is a set of Polish dialogues collected at the Warsaw Transport Authority call centre – the telephone service which provides information on tram and bus connections, schedules, routes, fares etc. The data set consists of 500 dialogues which were recorded in 2007, converted into texts and annotated with speech related facts. After the analysis of these data, a domain model consisting of attribute-value pairs was defined and used as a semantic tagset. The corpus annotated on several levels is the result of the LUNA FP6 EU project and was presented in [6] and [5]. It consists of 1385 turns containing 82977 occurrences of 5146 forms of 2759 lemmas. They are annotated with 20 chunk types, 206 concepts names and 47 frame types.

The subject of this paper is a structured version of the temporal part of the domain model elaborated for the task of corpus annotation. The set of labels was organized in the form of an ontology which contains all labels used to annotate time related notions in the corpus and introduces their multidimensional topology. Classes defined in the ontology represent temporal phrases which occurred in analyzed dialogues. These phrases are usually incomplete, imprecise or relative. In most cases they are disambiguated by human operators without asking any additional questions. It would be a desirable feature of any dialogue system if it could also cope with such imprecise time expressions. The first step to achieve this goal is to analyse time referring phrases which occur in natural spontaneous communication and define their typology. Then, the algorithm of resolving these incomplete references may be implemented.

In the rest of this paper we present the defined ontology of natural language time expressions together with examples of Polish phrases found within the collected dataset. The frequencies of all concepts in the entire corpus of dialogues are also analysed.

2 Temporal Expressions Ontology

The ontology presented in the paper is a newly defined OWL resource. There are two reasons for this solution. First, although a lot of effort has been put into making ontologies reusable, using already defined ones (like [8]) is still very difficult and all problems connected with their full understanding, projection into a chosen subdomain and expansion (issues described for example in [9]) still exist. The second reason was connected with our goal – making a specialized resource which describes only these temporal expressions which are used in a chosen domain and are formulated in a natural language which was not taken into account while building existing ontologies. Two main assumptions made while building the ontology was to introduce complex concepts like BEFORE_HOUR which include all time related modifiers within their limits (like in TIMEX2 not TIMEX3) and (also in opposition to TIMEX3) not to disambiguate forms like *dziś* 'today'. This task was left to the second stage of temporal data processing.

The defined time ontology represent s expressions which occur in real dialogues carried by users of public transport call center. It contains concepts divided according to three orthogonal criteria (see Figure 1):

- absolute vs. relative time descriptions,
- granularity level,
- time points vs. intervals.

According to the first criterion time notions are divided into: TIME_ABS – absolute descriptions (*o drugiej trzydzieści* 'at two thirty') and TIME_REL – relative descriptions (*w dniu dzisiejszym* 'today'). It should be noted that in the context of public transport information "absolute" time points, that is for example "8:30 21.01.2001", practically do not occur at all (there is no such expression in the entire corpus). A precise time description means in this context an hour within the day (it usually means 'today' or a type of a day, i.e. 'a weekday' or 'a holiday') or a date which is meant to be a datum within the current year or every year. The most frequent date in the corpus is September 1st (14 occurrences) which is the first day of a school year and is mentioned in the context of discounts.

The second division concerns two types of time points granularity. TIME_DAY_LEVEL class represents day-level descriptions (e.g. *31 sierpnia* 'August 31'), while TIME_HOUR_LEVEL class – descriptions at the level of hours and minutes (*trzecia trzydzieści, po południu* '3:30, in the afternoon').

The last criterion for time concepts division differentiate time points (TIME_DESC_POINT), time frequencies (TIME_RATE) and time intervals (TIME_DESC_INTERVAL). Time intervals are in turn divided into anchored time intervals (TIME_DESC_PERIOD) and time interval duration (TIME_SPAN).

Time related classes are presented in Table 1. Every class is described there by the number of phrases which it represents and (in the third column) by the number of its immediate subclasses or the number of different instances (for the lowest level of the hierarchy). In the first case, in the parentheses, the number

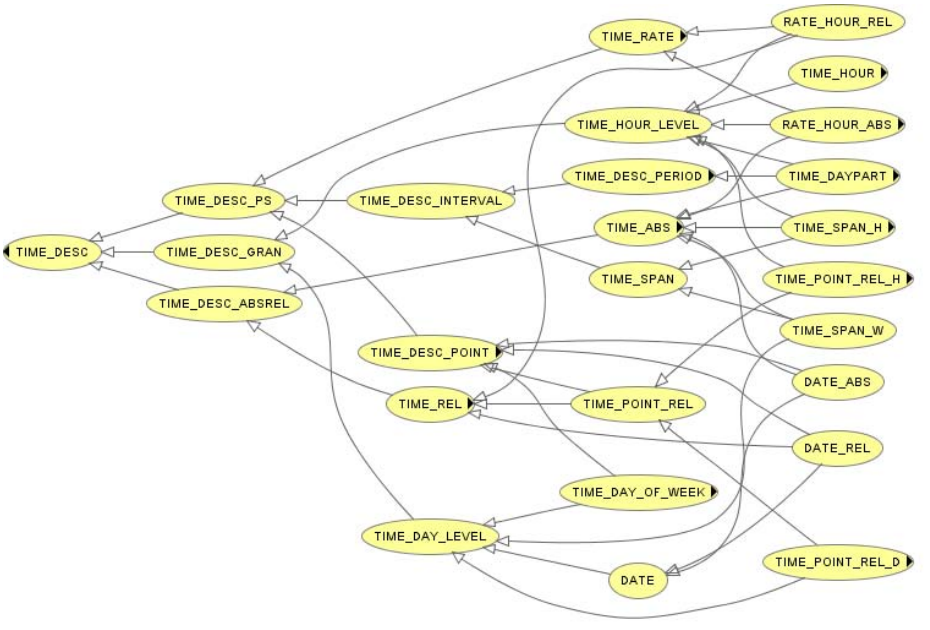


Fig. 1. Upper part of the class hierarchy

of all defined subclasses is given. The second case, in which a lot of class instances differing in values of a datatype property can occur, is marked by the '*'. The fourth column contains the number of different natural language phrases which occurred within the corpus. Their examples together with English translations (for cases in which a name of a class is not sufficient for understanding the phrase) are given in the last column. As it is not easy to clearly linearise the multidimensional division, the subsections which try to represent it are not always quite precise. For most classes, their exact placement in the hierarchy is given in Figure 1.

The numbers given support the statement that in this domain relative temporal expressions are very frequent. For example, for about 2300 time concept occurrences, there are 448 occurrences of TIME_REL objects which represent relative time of a day in intervals. On the other hand, the phrases used to represent them are not very diverse (41 type of phrases). The most frequent notion of this class is 'Later'. 170 occurrences were expressed by only 6 types of phrases. There are also quite a lot of partial expressions, not only those giving hours without giving minutes, but there are also a lot of expressions in which only the minute part is given (261). Such an expression can either mean an additional specification of an exact hour which was already given before, or address every hour in a given time period. The next observation is the fact that there are some domain related language expressions which are not likely to be found in general resources, e.g. *godziny szczytu 'peak hours'.*

Table 1. Temporal concepts statistics

name	occurr.	different subtypes /objects	different phrases	examples
TIME_DAY_LEVEL				
Points:				
DATE	18	2(2)	11	
DATE_ABS	14	8(*)	10	pierwszy września '1.09'
DATE_REL	4	2(*)	1	dzień urodzin 'birthday'
TIME_OF_WEEK_WRK	62	2(2)	20	
HOLIDAY	30	1	4	w dni nierobocze 'non working days'
WORKINGDAY	32	1	16	w dni robocze 'working days'
Intervals:				
TIME_YEARPERIOD_ABS	30	9(2)	11	
DATE_ABS_BEG	22	8(*)	10	od pierwszego kwietnia 'from 1.04'
DATE_ABS_END	2	1(*)	1	do 31 sierpnia 'to 31.08'
TIME_HOUR_LEVEL				
Points:				
AFTER_HOUR	33	19(*)	23	po dwudziestej 'after 20'
AROUND_HOUR	58	26(*)	57	około dziesiątej 'around ten'
AT_HOUR	757	420(*)	514	dziesiąta osiem 'eight past ten'
AT_HOUR_MINPART	261	62(*)	74	zero 'zero', piętnaście po 'fifteen after'
AT_HOUR_PART	38	23(*)	24	dwunasta '12:00'
BEFORE_HOUR	31	13(*)	15	przed dziesiątą 'before 10'
IN_X_HOURS	3	1(*)	3	za jakąś godzinkę
IN_X_MINUTES	16	8(*)	12	za jakieś pół godziny
Intervals:				
DAYTIME_PERIOD_BEG	50	21(*)	27	od godziny ósmej, od ósmej 'from 8:00'
DAYTIME_PERIOD_END	25	17(*)	21	do dwunastej 'till 12:00'
DAYTIME_PERIOD_SPAN	18	17(*)	18	między dziewiątą a dziesiątą
TIME_DAYPERIOD	79	8(8)	22	
_AFTERNOON	2	1	2	po południu, o tej porze południowej
_DAY	3	1	2	w ciagu dnia
_EVENING	4	1	4	w godzinach wieczornych, wieczorem
_MORNING	36	1	6	poranny, ranne godziny, rano, z rana
_NIGHT	12	1	1	w nocy
_OFFPEAK	1	1	1	poza szczytem
_PEAKHOURS	11	1	3	godziny szczytu, w szczycie
_WHOLEDAY	5	1	2	cały dzień
TIME_REL	448	11(11)	41	
_AtThisTime	12	1	3	o tej porze 'then'
_Earlier	23	1	7	wcześniej, wcześniejsze, przedtem
_Earliest	5	1	2	najwcześniej
_Now	95	1	4	teraz, w tej chwili
_Early	2	1	1	wcześnie
_Later	170	1	6	a potem i później tam
_TheSame	6	1	5	tą samą godzinę, tak samo
_Tomorrow	69	1	4	jutro
_Today	45	1	4	dziś, w dniu dzisiejszym
_TooEarly	6	1	2	(to) za wcześnie
_TooLate	45	1	2	(to) za późno
Span:				
TIME_SPAN	12	2(2)		
TIME_SPAN_D	3	2(*)	2	około tydzień 'around a week'
TIME_SPAN_H	9	8(*)	8	dziesięć minut 'ten minutes'
RIDE_DURATION_H	7	2(*)	2	mniej wiecej godzinę 'around an hour'
RIDE_DURATION_M	169	40(*)	92	maksimum czternaście minut
RIDE_DURATION_REL	11	2(*)	5	długo 'long'
WALK_DURATION	5	5(*)	5	dwie trzy minuty, minuta drogi dwie
Rates:				
RATE_HOUR_ABS	69	13(*)	28	co godzinę, raz na pól godziny
RATE_HOUR_REL	28	4(5)	12	
_Freq	10	1	4	często dosyć często,
_LessFreq	3	1	2	przerzedzenia, rzadziej
_MoreFreq	3	1	1	częściej
_NoFreq	12	1	6	dosyć rzadko, strasznie rzadko

3 Further Work

In the paper the ontology for representing Polish phrases referring to temporal expressions used in one chosen domain was presented. Two different directions of further usage of this resource are planned. First, the ontology will be utilised while implementing an algorithm for resolving time references in the original domain of public transport information. Second, the experiment with expanding this resource with time related notions which occur in a different domain of medical clinical notes is planned. The next possible subject for further research could be a detailed comparison of type of phrases used in Polish dialogues with phrases used in another natural language in a similar context. There is also planned to establish a method for automatic conversion of the elaborated annotation into TIMEX labels.

References

1. Ahn, D., van Rantwijk, J., de Rijke, M.: A cascaded machine learning approach to interpreting temporal expressions. In: HLT-NAACL, ACL, pp. 420–427 (2007)
2. Alexandersson, J., Reithinger, N., Maier, E.: Insights into the dialogue processing of VERBMOBIL. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, ANLP, pp. 33–40 (1997)
3. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: TIDES 2005 standard for the annotation of temporal expressions. Technical Report, MITRE (2005)
4. Mani, I., Pustejovsky, J., Gaizauskas, R. (eds.): The Language of Time. A Reader. Oxford University Press, Oxford (2005)
5. Mykowiecka, A., Głowińska, K., Rabiega-Wiśniewska, J.: Domain-related annotation of Polish spoken dialogue corpus LUNA.PL. In: Proceedings of LREC 2010 Conference (2010)
6. Mykowiecka, A., Marasek, K., Marciniak, M., Rabiega-Wiśniewska, J., Gubrynowicz, R.: Annotated corpus of Polish spoken dialogues. In: Vetulani, Z., Uszkoreit, H.(eds.) LTC 2007. LNCS, vol. 5603, pp. 50–62. Springer, Heidelberg (2009)
7. 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. 411–419. Springer, Heidelberg (2009)
8. Pan, F.: Representing Complex Temporal Phenomena for the Semantic Web and Natural Language. PhD thesis, Computer Science Department, University of Sothern California (2010)
9. Peralta, D.N., Pinto, H.S.: Reusing time ontology. In: Enterprise Information Systems V, pp. 241–248. Kluwer Academic Publishers, Dordrecht (2004)
10. Prior, A.: Time and Modality. Oxford University Press, Oxford (1957)
11. Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., Pustejovsky, J.: The tempeval challenge: identifying temporal relations in text. In: Language Resources and Evaluation, pp. 161–179 (2009)

Reliability of the Manual Segmentation of Pauses in Natural Speech

Raoul Oehmen, Kim Kirsner, and Nicolas Fay

University of Western Australia
Stirling Hwy, Crawley 6008, Perth, Western Australia
raoul.oehmen@graduate.uwa.edu.au

Abstract. Recent innovations regarding analysis of pauses in natural speech have necessitated the segmentation of increasingly small pause durations from the speech stream [1]. Identifying pauses and pause durations relies on human judgement. However the reliability of these judgements has yet to be established. This study investigated the reliability of multiple segmentations of four speech files. Results suggest that while inter-analyst reliability is moderate; intra-analyst reliability was high. Furthermore, inter-analyst variation appears to be related to the signal to noise ratio of the speech files. A further, analysis of the segmentation of one speech file demonstrated that a lack of reliability was associated with certain non-speech vocalizations, suggesting that reliability could potentially be increased with more precise guidelines for analysts.

Keywords: Pauses, Fluency, Distributional Fitting, Reliability.

1 Introduction

Traditional analyses of pause durations in spontaneous speech have applied arithmetic means to real-time pause frequency distributions [2] similar to those shown on the left in Figure 1. However, significant skew in these distributions makes this problematic [3]. More recently, the frequency of durations of pauses has been more accurately viewed in log-time where pauses form two normal distributions [1, 4], as shown on the right in Figure 1. These distributions can be mathematically modelled and descriptive statistics obtained. They have been dubbed the short-pause (SP) and long-pause (LP) distributions and are thought to represent pauses related to articulatory and cognitive processes respectively [1], in line with traditional accounts [2].

However, prior to modelling pause frequency distributions, pauses must first be identified from the speech stream. For more than forty years, a widespread disinterest in short, articulatory pauses (due in part to poor recording and analysis resolutions) has led to the use of a variety of minimum pause duration thresholds (the smallest pauses considered to be 'real'). Numerous different thresholds used across different studies have made comparisons difficult as changes in threshold affects almost every descriptive statistic [5]. The pervasive 250msec threshold employed by Goldman-Eisler [2] to exclude articulatory pauses was perhaps inaccurate as numerous pauses between 130 and 250msec have been shown to have both cognitive and expressive

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 263–268, 2010.

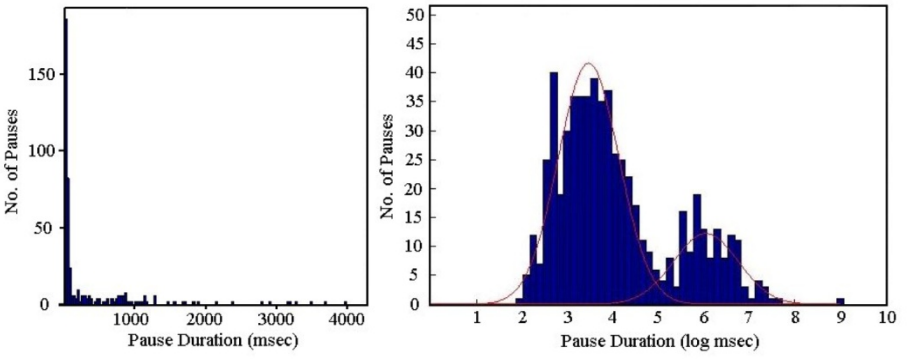


Fig 1. Pause frequency distributions in both real-time (left) and log-time (right) for a typical speaker. Note: log-time frequency distribution includes bimodal two distributional fit.

functions [6]. Furthermore, the junction between articulatory and cognitive pauses has been shown to be highly variable between individuals [1].

Technological and theoretical advances have allowed more recent approaches to embrace articulatory pauses via the use of ultra short minimum pause thresholds (no minimum [7], 10 msec [4] and 20msec [1]). This raises the additional question of the accuracy of such fine judgements by analysts in discerning vocalisations from brief silences.

Such judgements are necessary regardless of whether pauses are segmented from speech entirely by hand or whether a semi-automated procedure is used (e.g. [8] and [9]). Semi-automated procedures typically consider speech to be any time when an amplitude contour rises above some threshold: usually set manually to the level of ambient noise [10]. Thus, manual and semi-automated procedures differ only with regards to whether analysts pick a single 'best-fit' threshold or decide upon thresholds on a pause by pause basis.

While semi-automated systems have an advantage in terms of consistency [6], their performance may be degraded under conditions of low signal to noise ratios caused by a range of factors including recording conditions, stress and aphasia. Furthermore, while automated systems are reliant purely on the amplitude contour, modern analysis equipment puts many more measures at the disposal of the manual analyst with none of the resolution problems inherent in early studies [2]. Albeit with a few exceptions [4, 11], reliability statistics have seldom been reported and no detailed analysis of variations between segmentations (regardless of measurement techniques) has been conducted to date. This study will determine both Inter-analyst and Intra-analyst reliability in addition to investigating the causes of any observed variation.

2 Method

2.1 Stimuli

The stimuli to be analysed consisted of four spontaneous monologues elicited via a stimulus question (averaging 1.86 minutes and containing on average 162 discernable pauses). Each contained a different English first language, adult speaker (3 females

and 1 male) and each discussed different topics. Files MB, SK & CN were recorded on dynamic, unidirectional, lapel microphones into portable, digital recorders. File RO was recorded using a head-mounted, dynamic, unidirectional microphone in an acoustically controlled laboratory, with a signal to noise (S/N) ratio of 53.86. Files SK & CN were recorded in a quiet laboratory setting and had S/N ratios of 20.36 and 16.36 respectively while File MB was recorded in a home setting and had a S/N ratio of 5.90. All files were recorded at a sampling frequency of 44100 Hz.

2.2 Segmentation

Four experienced pause analysts (two qualified Speech Therapists, one Linguist and one Psychologist) segmented each of the sound files. In addition, each analyst segmented one of the four files a second time after an interval of roughly one week. This provided five segmentations of each sound file and 20 segmentations in total. Segmentation was conducted in PRAAT [12] via the procedure described in [13]. This 'physical' approach ignores linguistic considerations entirely, requiring analysts only to place boundaries at those points in the file where a period of silence transitions into a period of speech and vice versa. Thus each pause consists of two boundaries, acknowledging the on-off sequence of vocalisations that comprise speech. Sounds that were clearly not associated with a speech act (such as non-stylistic coughs and audible movements of the muscles of articulation) were excluded. Analysis was conducted primarily via the visual modality (making use of spectrographic information including amplitude and fundamental frequency contours) but also by listening to the speech. The minimum pause duration threshold employed in the present study was 10msec while the viewing window was set at 0.6 seconds.

2.3 Distributional Analysis

Pauses were modelled in an analogous fashion to that reported in [1]. Pause durations were converted to logarithms and grouped into a series of bins forming pause frequency distributions for each speaker separately. A Expectancy-Maximisation algorithm [14] was then applied to each distribution fitting the best two-distribution model to the data in an attempt to minimise log-likelihood. A number of statistics can then be derived from the modelled distributions including the means, standard deviations and pause rates (per minute of speech) of the 'long' and 'short' pause distributions.

3 Results

Distributional statistics for each segmentation were obtained following application of the Expectancy-Maximisation algorithm [14]. Measures of variation were calculated for Inter-analyst reliability (between the different segmentations of the same file by different analysts) by calculating the standard deviation of the four segmentations of each file, for the Long and Short Pause distributional means. Similarly, measures of Intra-analyst variation were calculated as the standard deviation of the repeat segmentations of one file by each analyst. The measures are presented in Table 1 below, and show intra-analyst variability to be lower than Inter-analyst variability. Furthermore, the data indicate that variation increases with signal to noise ratio. This point is seen in Figure 2, where larger shaded areas (representing variation between segmentations)

are associated with those files with lower signal to noise ratios. This is further demonstrated by a Pearson correlation coefficient of r = -0.60 between the variability in short pause means and signal to noise ratio, albeit with only 4 data points.

Table 1. Signal to noise ratio and Inter/Intra analyst distributional statistic variation (for the short and long pause distribution mean) of four speech files in log

		File MB	File CN	File SK	File RO
	Signal/Noise Ratio	5.90	16.36	20.36	53.86
Inter	SP Mean: Variation.	0.50	0.16	0.49	0.17
	LP Mean: Variation	0.32	0.09	0.13	0.07
Intra	SP Mean: Variation	0.18	0.01	0.02	0.03
	LP Mean: Variation	0.05	0.04	0.03	0.00

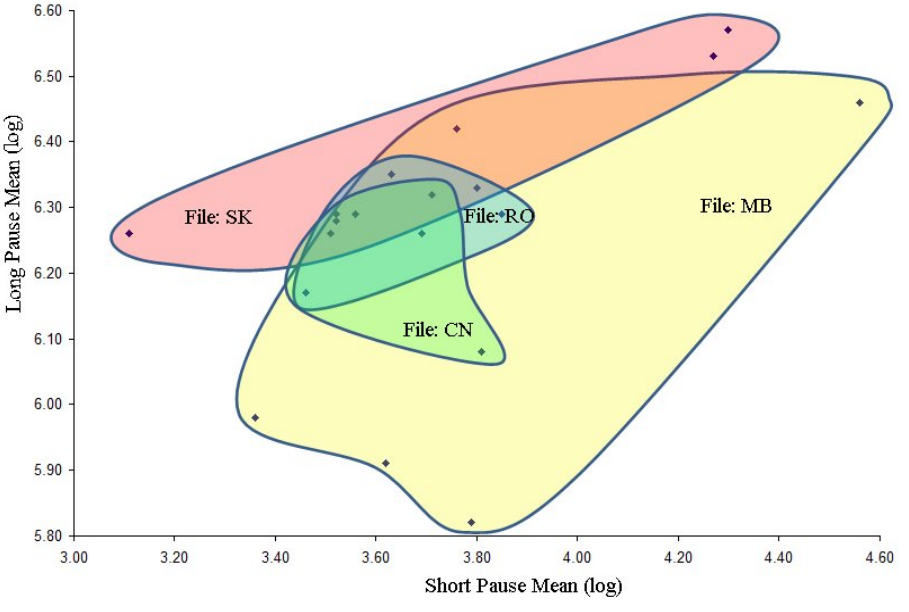


Fig. 2. Long pause mean plotted against short pause mean for all segmentations of four speech files (inter-analyst reliability). Shaded areas represent the spread of distributional parameters between analysts' segmentation of the same file.

Despite File RO being the highest quality recording, variation still remained between analysts. A more detailed analysis of the placement of boundaries (junctions between speech and pause) between segmentations of this file was therefore conducted. Two important analyses were conducted: the reliability with which analysts find a particular junction (% agreement) and the variability (in msec) of the positioning of boundaries between analysts. Variation between positioning of analysts' boundaries was calculated by taking the absolute root mean square difference as used in [4]. For the present file

this was 5.83msec for all boundaries located by two or more analysts. For voice offset boundaries (speech changing to pause), those followed by a short pause had significantly less variation compared to those followed by a long pause (absolute root mean square difference of 4.41msec and 12.63 respectively, $t(61) = -2.65$, $p < 0.01$.

With regards to agreement, out of a total of 471 boundaries discovered by one or more analysts 66.03% of boundaries were located by all four analysts, while 76.92% of boundaries were located by at least 3 of the 4 analysts. Only 14.32% and 8.76% of boundaries were located by one or two analysts. In an attempt to determine more specifically the cases of low agreement, all speech surrounding boundaries was coded into phonemic categories. Table 2 below, shows the proportions of some of these events for both speech onset and speech offset boundaries.

Boundaries located by one or two analysts appear to be characterised by a disproportionately high percentage associated with preparatory movements for articulation (e.g. sounds emitted by opening the mouth prior to vocalisation) as well as boundaries associated with fricatives. Similarly, there appears to be a disproportionately low number of boundaries associated with plosives that were detected by only one analyst.

Table 2. Percentage of boundaries comprising each phonemic category at varying agreement levels separated by whether the boundary signifies the onset or offset of speech

Category	Onset of Speech (after pause)				Offset of Speech (before pause)			
	1/4	2/4	3/4	4/4	1/4	2/4	3/4	4/4
Nasal	0.00	4.76	0.00	1.96	14.71	5.00	12.00	23.37
Plosive	12.12	33.33	42.31	41.18	11.76	25.00	28.00	23.38
Fricative	24.24	19.05	15.38	18.30	23.53	10.00	8.00	16.23
Long Vowel	9.09	4.76	3.85	7.19	8.88	35.00	12.00	11.69
Short Vowel	12.12	9.52	7.69	22.88	11.76	0.00	16.00	12.34
Prep. Artic.	21.21	14.28	19.23	1.31	25.71	5.00	12.00	2.60

4 Discussion

Results suggest that while intra-analyst reliability is high; inter-analyst reliability is only moderate with sizable differences between segmentations that can be linked to the signal to noise ratio of the files. With regards to intra-analyst segmentations, the data suggest that whatever reasoning or judgement is being used, it is being applied consistently. However, in inter-analyst segmentations, an increase in variability at low S/N ratios could be due to a decrease in the discriminability of the signal. This would bring differences in analysts' criterion (or their propensity to say the signal was present) to the fore; in line with traditional Signal Detection Theory accounts [15]. While these findings suggest that attention should be paid to the quality of recordings (as well as using only a single analyst where possible); it is clearly not feasible to insist on laboratory quality recordings. Instead, future research could investigate possible increases in accuracy associated with the use of additional variables (e.g. pulse rates and formant frequency) as well as changing spectrogram viewing settings.

A further, detailed analysis of the positioning of boundaries in a single file revealed greater variation related to the positioning of boundaries proceeding long pauses (pauses occurring after the completion of a motor plan), compared to proceeding short

pauses (pauses occurring within a motor plan). One possible explanation is that the occlusion of vocalisation within a motor plan is likely to be under stricter temporal constraints due to the following speech than at the completion of a motor plan. This may lead to a more defined boundary and make analysis more precise. Nonetheless, variation in the placement of boundaries does not account for the extent of the variation between analysts' statistics. Variation is more likely to be due to imperfect agreement on the presence or absence of boundaries. In particular, it appears that disagreements are frequently caused by misclassification of non-speech artefacts such as a preparatory articulation and audible breathing events as speech. Such problems can likely be resolved with improved segmentation guidelines for analysts.

References

1. Kirsner, K., Dunn, J., Hird, K., Parkin, T., Clark, C.: Time for a pause... Speech Science Technology, Melbourne (2002)
2. Goldman-Eisler, F.: Psycholinguistics: Experiments in spontaneous speech. Academic Press, London (1968)
3. Quinting, G.: Hesitation phenomena in adult aphasic and normal speech. Mouton, The Hague (1971)
4. Rosen, K.M., Kent, R.D., Duffy, J.R.: Lognormal distribution of pause length in ataxic dysarthia. Clinical Linguistics and Phonetics 17, 469–486 (2003)
5. Kowal, S., Wiese, R., O'Connell, D.C.: The use of time in storytelling. Language and Speech 26, 377–392 (1983)
6. Hieke, A.E., Kowal, S., O'Connell, D.C.: The trouble with "Articulatory" pauses. Language and Speech 26, 203–214 (1983)
7. Campione, E., Veronis, J.: A large-scale multilingual study of silent pause duration. Speech Prosody, Aixen-Provence (2002)
8. Jaffe, J., Feldstein, S.: Rhythms of dialogue. Academic Press, New York (1970)
9. Greene, J.O., Cappella, J.N.: Cognition and talk: The relationship of semantic units to temporal patterns of fluency in spontaneous speech. Language and Speech 29, 141–157 (1986)
10. Glukhov, A.A.: Statistical analysis of speech pauses for Romance and Germanic languages. Soviet Physics and Acoustics 21, 71–72 (1974)
11. Greene, J.O., Lindsey, A.E., Hawn, J.J.: Social goals and speech production: Effects of multiple goals on pausal phenomena. Journal of Language and Social Psychology 9, 119–134 (1990)
12. Boersma, P., Weenink, D.: PRAAT: Doing phonetics by computer. Institute of Phonetic Sciences: University of Amsterdam, Amsterdam (2005)
13. Kirsner, K., Dunn, J., Hird, K.: The long and short of pauses in natural speech: Analysis and application to aphasia (in preparation)
14. McLachlan, G., Peel, D.: Finite Mixture Models. John Wiley & Sons, Canada (2000)
15. Wickens, T.D.: Elementary Signal Detection Theory. Oxford University Press, New York (2002)

Large-Scale Language Modeling with Random Forests for Mandarin Chinese Speech-to-Text

Ilya Oparin, Lori Lamel, and Jean-Luc Gauvain

LIMSI CNRS, Spoken Language Processing Group,
B.P. 133, 91403 Orsay cedex, France
{oparin,lamel,gauvain}@limsi.fr
http://www.limsi.fr

Abstract. In this work the random forest language modeling approach is applied with the aim of improving the performance of the LIMSI, highly competitive, Mandarin Chinese speech-to-text system. The experimental setup is that of the GALE Phase 4 evaluation. This setup is characterized by a large amount of available language model training data (over 3.2 billion segmented words). A conventional unpruned 4-gram language model with a vocabulary of 56K words serves as a baseline that is challenging to improve upon. However moderate perplexity and CER improvements over this model were obtained with a random forest language model. Different random forest training strategies were explored so as to attain the maximal gain in performance and *Forest of Random Forest* language modeling scheme is introduced.

Keywords: language modeling, random forest, speech-to-text, ASR, STT, Mandarin Chinese.

1 Introduction

The task of language modeling is to create language models (LM) that are able to capture the regularities of a natural language. A language model is an inherent part of any modern speech-to-text (STT) system. Language models are used in STT systems along with acoustic models (AM) and a pronunciation dictionary that links the them, to perform large vocabulary continuous speech recognition. The basis of modern language models is the word *N-gram* approach. The omnipresent assumption of N-grams is that the occurrence of a word can be predicted according to its immediate (in case of bigrams) or short-range left context. Word N-grams appear extremely efficient in practice. Frankly speaking, despite the fact that it has already been many decades since they were introduced into the recognition field, N-grams are still the most common framework for language modeling, since their modeling capacity is hard to beat.

This work aims to improve over a baseline N-gram model by using random forest LMs. The peculiarity of this work is that the brute-force N-gram LM is trained on very large amounts of text data (over 3 billion of word tokens)

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 269–280, 2010.

without any pruning and cut-off. This model is thus robust and very challenging to improve upon. The baseline recognition system is a competitive state-of-the-art Mandarin STT system that was developed at LIMSI for and submitted to the GALE Phase 4 evaluation.

Language models are usually evaluated by means of word error rate (WER) and perplexity. Due to peculiarities of Mandarin Chinese, it is common practice to measure speech recognition performance in terms of the character error rate (CER) rather than the traditional WER. However, WER or CER do not give an opportunity to compare LMs directly, since WER/CER results also include the impacts of both the acoustic model and the decoder. Thus if the systems to be compared differ either in the acoustic component or in the decoder, no LM comparison can be made. The *perplexity* is widely used in speech recognition community to evaluate the LM independently from the full system:

$$PP = 2^{\hat{H}} = P_M(w_1, w_2 \dots w_m)^{-\frac{1}{m}} \tag{1}$$

Where $\hat{H}$ is a per-word entropy, $P_M()$ is the probability assigned to a string of words from a test corpus by a LM and m is the length of a string in words.

In the next section random forest approach (RF) to language modeling is introduced. The data and experimental setup are described in Section 3. Perplexity and speech recognition accuracy results are given in Section 4, with conclusions and directions for future work presented in Section 5.

2 Random Forest Language Models

2.1 Decision Trees

The decision tree (DT) mechanism for estimating probabilities of words following each other has long been known as an alternative to the N-gram approach for language modeling in STT [1]. Several studies showed that stand-alone DTs do not outperform traditional smoothed N-gram models [2]. However, with recent advances in language modeling that extend the use of decision trees to that of random forests, this research direction has reentered the research spotlight.

With the help of DTs it is possible to cluster together similar histories (i.e. possible previous words to the one being predicted) at the leaves of a tree. Each leaf forms an equivalence class of histories that share the same probability distribution over words to predict. Usually binary DTs are implemented in which sets of possible histories are split at every node with a *yes/no* question. If the predictor (i.e. position in N-gram history we ask questions about) is the previous word, a question looks like "Is the previous word in the set S or $\overline{S}$?" The data (i.e. N-grams) corresponding to *yes* answers are propagated through one branch going out of a node, the *no*-data is passed along the other branch. Actually, a conventional N-gram model can be represented as a special case of a tree model. For example a bigram model may represented by a DT in which the *yes* set consists of one individual word at each node. Ideally, during the training phase all possible predictors and questions should be tried at each node to split the

data and the "best" predictor/question pair should be picked and stored for that node. However, in real life greedy algorithms have to be used.

A DT is constructed in a way to reduce the uncertainty about the event being predicted. Thus, entropy can naturally be used as the goodness measure. One should measure entropy for training data M in a node before split, then split data in two sets S and $\bar{S}$ according to *yes/no* questions and find the entropy reduction under the split. Minimizing entropy is equivalent to maximizing the log-likelihood, where the log-likelihood under the split is formulated as

$$\begin{aligned} L(S) &= \sum_w \left[C(w,S) \log \tfrac{C(w,S)}{C(S)} + C(w,\bar{S}) \log \tfrac{C(w,\bar{S})}{C(\bar{S})} \right] \\ &= \sum_w \left[C(w,S) \log C(w,S) + C(w,\bar{S}) \log C(w,\bar{S}) \right] \\ &- C(S) \log C(S) - C(\bar{S}) \log C(\bar{S}) \end{aligned} \tag{2}$$

and $C()$ is a count of an event. The increase in log-likelihood on the training data shows how good the question is. Different questions (and thus different splits) should be tried and the one with the best log-likelihood is picked for a given node. Stopping criterion should also be introduced at the stage of tree-growing. If no stopping criterion is used, the resulting DT is grown until there is one N-gram at each leaf. The log-likelihood of a DT can always be increased by increasing the number of leaves. However, such a tree will not be able to generalize well on unseen data. Stopping criteria are therefore used to ensure reasonable termination of branching. There are a number of possible criteria, for example a minimum log-likelihood increase threshold. However, these constants are empirical and call for manual tuning. An alternative is to measure the log-likelihood of heldout data under the same split as for the training data. To do so, DTs must be grown on training data, the log-likelihood checked on heldout data, and all nodes that with no increase in log-likelihood deleted.

When traversing a DT, the data is split in two subsets at each node that causes data sparsity. Each leaf of a tree contains a probability distribution over all the words in the recognition vocabulary. It is possible that some N-grams propagated to a certain leaf will get zero probability. Thus, just as standard N-gram probabilities, DT probabilities should be smoothed. This can be done via the discounting and smoothing techniques developed for language modeling or by means of recursive in-tree smoothing with parent node probabilities as proposed in [3].

The complete mathematical formulations for the issues described above are not presented due to lack of space. The reader is referred to [4] for the the details concerning DT-based language modeling.

2.2 Random Forests

A random forest is a collection of decision trees that include randomization in the tree-growing algorithm. The underlying assumption is that while one DT does not generalize well to unseen data, a set of randomized DTs interpolated

together might and actually should perform better. Greedy algorithms are used at the stage of DT construction for choosing the best questions to split the data. We also do not take into account questions asked at other nodes when searching for the one to be asked at a given node. As a result, trees are only locally, but not globally, optimal (with respect to training data). Randomized trees where the randomization is introduced during tree construction (i.e. finding the "best" questions to ask at each node) are not locally optimal, but the collection of them may be - and actually is - closer to the global optimum and thus these provide better results. Different randomization schemes may be used to randomize DTs in order to form a RF. The most commonly used methods are random predictor selection to ask questions about and random initialization of greedy algorithms used to find the "best" question in a node.

It should also be noted that the RF approach is also a promising framework to incorporate different sources of information into a language model [4,6].

The RF models were shown to consistently outperform word-based N-gram models for relatively small-scale tasks (e.g. the Wall Street Journal portion of Penn Treebank) [2,4,5]. [7] reported improvements in recognition performance with random forest LMs (trained on limited data) that take account of morphological features for inflectional languages. Improvements in recognition rate after rescoring N-best lists (generated with a conventional N-gram model) with a RF model were also reported for Mandarin Chinese for the GALE task with LMs trained on about 700 million word tokens of data [5,8]. The setup used here is similar to the latter. However our experiments are characterized by significantly larger training data size and a lower CER for the recognition baseline.

3 Experiments

3.1 Chinese Mandarin STT System

The GALE Phase 4 Mandarin Chinese setup was used for the current experiments. The system is a highly competitive one with a language model trained on large amounts of Mandarin Chinese data, thus providing the system with robust linguistic estimates. This makes improving upon the performance attained with this system a very challenging task.

Recognition vocabulary. In written Chinese, words are not separated by white spaces. The natural solution is thus to make use of character-based LMs or perform word segmentation as a pre-processing step. The former was shown to be inferior to the latter [9], so the segmentation approach was taken in this work. Due to ambiguity word segmentation in Chinese is not a trivial task as even native speakers of Chinese may disagree in certain cases [10]. Several approaches to automatic word segmentation in Chinese exist [11]. In this work we make use of the longest-match algorithm based on the 56052 word vocabulary used in LIMSI STT systems developed for previous GALE Mandarin Chinese evaluations. This is a simple greedy algorithm that tries to match the longest possible word according to the vocabulary, adds a space after it and shifts to the

next character after the space to search for another word. The segmented files are used to train word-based LMs. The vocabulary also includes all individual Chinese characters. This way we avoid the problem of having out-of-vocabulary words in a text after the segmentation.

Decoding. The speech recognizer is a development version of the LIMSI STT system used in the AGILE participation in the GALE'09 evaluation. Word recognition has one decoding chain with three passes. The first pass generates a word lattice with cross-word, position-dependent, gender-independent acoustic models, followed by consensus decoding with 4-gram and pronunciation probabilities [12,13]. Unsupervised acoustic model (AM) adaptation is performed for each segment cluster using the CMLLR (Constrained Maximum Likelihood Linear Regression) and MLLR [14] techniques prior to the next decoding pass. The first decoding pass is done with an MLP+PLP+f0 acoustic model, the second uses a PLP+F0 based model, and the third pass also uses an MLP+PLP+f0 acoustic model.

All AMs are tied-state, left-to-right context-dependent (CD), HMMs with Gaussian mixtures. The triphone-based CD phone models are word-independent but position-dependent. The tied states are obtained by means of a decision tree. The models all use speaker-adaptive (SAT) and Maximum Mutual Information Estimation (MMIE) training. They are trained on 1400 hours of manually transcribed broadcast news and broadcast conversation data distributed by LDC for use in the GALE program, using both standard PLP and concatenated MLP+PLP features. For the PLP models, a maximum-likelihood linear transform (MLLT) is also used. The model sets cover about 49k phone contexts, with 11.5k tied states and 32 Gaussians per state. Silence is modeled by a single state with 2048 Gaussians. Initially speaker-independent models are trained on all of the available data, and serve priors for Maximum a Posteriori (MAP) estimation of gender-specific models.

LM training data. The LM training data consists of 48 different text sources in Mandarin Chinese. These sources are collected by different institutions and are diverse in size, genre and internal structure. The data includes transcripts of broadcast news and broadcast conversations, newspaper texts, text collected from the web, etc. The description of the data available for the GALE Phase 3 evaluation can be found e.g. in [15]. For the Phase 4 evaluation new text corpora of Mandarin Chinese became available. Some corpora are entirely new, some are the extended versions of the ones that existed before. The new data were added to that used for the previous evaluations to train the language models.

The total amount of new data is 590.32M words after segmentation. It represents a 22.5% increase in the data available for training as compared to the data available for the previous evaluations (2.6G words), resulting in a corpus with 3.2 billion word tokens.

Baseline Language Model. The baseline LM is a word-based 4-gram LM. Individual LMs are first built for each of the 48 corpora. These models are smoothed

Table 1. Perplexity on different GALE evaluation sets

Set	*Phase 3 LM*	*Phase 4 LM*
dev07	181	184
eval07	206	206
dev08	194	192
dev07+eval07+dev08	193	194
dev09	234	211
dev09s	230	207

according to unmodified interpolated Kneser-Ney discount scheme [16]. No cut-offs and pruning is imposed thus allowing the LMs to take account of all possible information. These individual models are subsequently linearly interpolated together with the interpolation weights tuned on *dev09* data. It should be noted that the dev data is never used it for RF tuning (e.g. as held out data for controlling the DT growing process). This is done in order to keep the same test conditions and to not introduce any biases. As the number of individual models is 48 (one model is trained for each available corpora), this small number of parameters does not result is bias towards this data. This is supported by the comparison with the previous baseline 4-gram model developed for the GALE Phase 3 evaluation. That LM was tuned on the *dev07+eval07+dev08* subset. As can be seen from Table 1 current Phase 4 LM attains the same perplexity results on *dev07+eval07+dev08* data even though it was not tuned on these and it performs significantly better on *dev09* and *dev09s* data.

The GALE Phase 4 *dev09* sets were used in this study to evaluate the performance of different models. A subset of *dev09* called *dev09s* was also defined for this evaluation. It constitutes about half of *dev09* data. The baseline N-gram Phase 4 LM perplexity is 211 on *dev09* and 207 on *dev09s* sets.

In the system submitted to Phase 4 evaluation word lattices generated with the baseline LM are subsequently rescored with the Neural Network language model (NNLM) [17]. However, in this study the NNLM was not applied in order to assess the improvement with RFLMs over N-gram models.

3.2 Training of RF Models

The SRILM-compatible RF toolkit was used in these experiments with random forests [18].

Growing DTs for large training data and large vocabularies is very computationally expensive. We found it infeasible to do straightforward RF training for the 3.2 billion words data used to train the baseline N-gram model. Thus in our experiments we tried different strategies to train RF models that would result in building RFLMs in reasonable time, benefit from all the available data and improve the baseline at the same time.

An important feature of tree-based models is that after an RF is grown, i.e. the structure of constituent randomized DTs are established it is possible to

Table 2. Data chosen to estimate RF probabilities

corpus	*word #*	*sampling rate*	*sampled word #*
bcm/bnm data	19.42M	1.0	19.42M
ng	315.52M	0.01	3.15M
giga_xin	366.96M	0.008	2.93M
ibm_sina	279.87M	0.01	2.79M
giga_cns	76.73M	0.03	2.28M

pour larger amounts of data down to the leaves. The structure of a DT is a set of nodes and leaves together with questions that are assigned to each node. A question for a binary word-based DT is a position in history it is asked about (e.g. 1 for an immediate left neighbor) and words constituting *yes/no* sets for this position. Thus after a DT is grown we can propagate data down to the leaves: if a particular N-gram contains a word from a *yes* set at a specific position, it is propagated along one branch, if it contains a word from the *no* set the N-gram is pushed along the other. If a N-gram contains a word that is in neither set, the background N-gram LM is used to bailout to estimate the probability. Thus any N-gram either ends up in the leaf or gets its probability from the backoff LM. Pouring larger amounts of training data down to the leaves make probability estimations more robust since they are based on larger data.

RF on restricted data. Decision tree training may be performed on restricted data. This is the first experiment we ran before addressing the problem of making use of all available data. The crucial point is thus to choose the training and heldout data that is likely to be representative of the test data. For the GALE Mandarin Chinese task this is broadcast news Mandarin (bnm) and broadcast conversations Mandarin (bcm) transcribed data as it constitutes the target type of data in the evaluations. The training data was chosen to contain all available *bnm* and *bcm* transcriptions except for the recent *bcm* and *bnm* data released during Phase 4. The latter was chosen as the heldout data used as a stopping criterion during the DT training phase.

After the structures of constituent DTs are defined, the training data together with additional data is poured down to the leaves to get more robust probability estimates. The additional data was taken from the remaining top four (according to the interpolation weights) text sources. These text sources are quite large and thus were downsampled. The resulting size corresponds to the weights inferred during interpolation of separate source N-gram LMs to form the baseline N-gram model (see Table 2).

Heldout data is usually used as a stopping criterion during the DT training phase. In the SRILM-compatible RFLM toolkit [18] the DTs are actually fully grown on the basis of the training data and then pruned according to gains on heldout data. However in [8] it was shown that shallow RFs that contain DTs of limited "depth" have performance close to the RFs consisting of "fully grown" DTs. We thus first compare the performance of RFs consisting of fully grown

and shallow DTs. Another issue that needs evaluation is the number of DTs to form an RF. Usually 100 or 50 randomized DTs are sufficient to train a RF. The perplexity results for different RF configurations are presented in Table 3. The numbers 50 and 100 correspond to the number of randomized DTs that constitute a RF. The second and third columns correspond to the performance of RFs as stand-alone models while the last two columns show the perplexity when the RFs are interpolated with the baseline 4-gram LM.

As can be seen from this table while the RFs with DTs of maximum 1000 nodes appear to be too shallow, the ones with 10000 nodes perform close to the fully grown (and subsequently pruned) trees. There is also no really significant difference between RFs consisting of 50 and 100 trees trained on restricted data.

RF trained on different sources. As already mentioned, the baseline 4-gram LM is obtained as result of interpolation of many sub-LMs each being trained on one of 48 available Mandarin Chinese corpora. The interpolation weights are tuned on *dev09* data. Applying the same strategy to RF construction seems a natural thing to do. A problem arose in that the corpora are very large, containing hundreds of millions words which was found infeasible to train RFs straightforwardly. Thus a different strategy was utilized for these corpora. Individual RFs are not trained for large corpora. The DTs trained on restricted data are used instead. The data of large corpora is poured down these trees. Thus tree structures are the same as for the restricted data RF but the probability distributions in the leaves are estimated on the data from specific large corpora.

There are a total of 48 sources used to train Mandarin Chinese models. It was found to be feasible to train specific RFs for 34 of the sources. Randomized DTs (or, to be more precise, their structures in terms of nodes and questions asked in the nodes) trained on restricted data (see section 3.2) were used to pour down the counts for the larger 14 corpora. Training RFs for specific sources consumes a lot of computational time and puts high demand on memory usage. Training about 50 full-grown RFs consisting of hundred DTs may keep busy a modern computational cluster with couple of dozens nodes for months. The performance of shallow trees with maximum 10000 nodes was shown to be close to that of the full-grown trees on restricted data. At the same time such trees are much faster to train. As a result we trained shallow DTs with a maximum of 10000 nodes for each individual corpora. Another decision that was taken on the basis of these results is that 50 randomized DTs are basically enough to form a RF.

The final RF is obtained by interpolation of RFs corresponding to different sources. We call such an RF a *Forest of Random Forests* (FRF).

Table 3. Dev set perplexity for different RF configurations on dev09 set

DT depth	*50 DTs*	*100 DTs*	*50 DTs interp*	*100 DTs interp*
fully grown	279.4	276.1	206.8	206.4
10000 nodes	299.1	295.7	207.9	207.7
1000 nodes	358.1	356.1	210.7	210.7

4 Results

Decision tree probabilities were discounted according to modified Kneser-Ney scheme and used together with a corresponding Kneser-Ney smoothed 4-gram LM. The N-gram LM is used as a backoff model. This backoff model is trained on restricted data (see section 3.2). A total of 50 randomized DTs form the random forest.

As already mentioned, the perplexity of the *dev09* and *dev09s* data sets are respectively 211 and 207 with the baseline 4-gram LM. For *dev09s* set, the perplexity with the RF trained on restricted data is 293, which is higher than that with the best interpolated N-gram LM trained on all available data. When these two are interpolated together the perplexity of this data set decreases to 201, corresponding to a 3% relative improvement. However we are mostly interested in checking the results on *dev09* since this set was used to tune the N-gram models for individual corpora in order to form the baseline N-gram LM.

The *dev09* perplexity with different RFLMs are given in Table 4. The random forests were trained on different sources as described in Section 3.2.

The RF type *RF* corresponds to the RF trained on restricted data as described in Section 3.2. As for the *dev09s* setup, the RF on its own performs worse than the N-gram LM, but a small gain in perplexity is observed when these models are interpolated. According to results obtained with RFLMs using smaller setups one would expect a perplexity reduction over N-gram baseline with standalone RF models if trained on the same data. However, in these restricted data experiments we by definition use much less data to train a RF and also do not make use of interpolation of LMs trained on the different data sources.

The *FRF* in Table 4 stands for the Forest of Random Forests that takes account of all available data with interpolation weights tuned on *dev09*. The stand-alone perplexity of the FRF is 10% better than the perplexity of the RF. However, no further improvement after the interpolation with the baseline N-gram model was observed.

The RF minimum and maximum perplexities for individual corpora RFs are 302 and 414. The perplexities with the different individual N-gram models range from 485 to 2614. The perplexity distribution for individual corpora RFLMs are thus much flatter. This must be due to the fact a backoff N-gram LM plays significant role in RF probability estimation. Another explanation of this fact is the shallow nature of randomized DTs that form RFs with the limitation of 10000 nodes that was imposed on the models. This makes individual models "smoother" and as a result the interpolation of these models is actually less promising as compared to that of individual N-gram LM models.

Having observed the interpolation weights for different RFs in FRF, we found them considerably different from the weights of N-gram models that form the final N-gram LM. In FRF the RFs for the smallest corpora obtained unexpectedly large weights. We presume the reason is as follows. The modified Kneser-Ney N-gram LM trained on restricted data is used as a backoff model for the RF. The Kneser-Ney discount coefficients for 4-grams are rather high, especially for the singleton 4-grams (discount1 0.873236; discount2 1.476549; discount3+

Table 4. Perplexity for differently trained RFLMs on dev09 set

RF type	*Stand-alone ppl*	*Interpolated ppl*
RF	299	207
FRF	268	208
FRF with N-gram weights	283	210
FRF without small corpora	279	208
RF modified KN discounts	334	208
FRF modified KN discounts	282	208

1.364567). Thus, a lot of probability mass is taken from RF estimates and transferred to the N-gram model that is used as a backoff model. This effect is more severe in case an RF is trained on very small corpora (in our setup there are four corpora that are several order of magnitude smaller than the others) and thus contain many singleton 4-grams. The smallest corpora got as much as 0.3 of the total weight, which looks rather strange. The backoff N-gram LM thus contributes most to the final probability estimation. At the same time the N-gram backoff model is trained on restricted data performs well with the perplexity of 309 on *dev09*. Consequently at the stage of weight optimization for different RFs, these models may be given prominence. This helps to reduce the perplexity of the stand-alone FRF but not in interpolation with the baseline N-gram LM. In order to compensate for this effect we tried two experiments. First, we did not try to optimize weights for individual RFs but rather took the interpolation weights that were calculated to build the baseline N-gram LM from N-gram LMs corresponding to each of 48 corpora. The results given in the row in *FRF with N-gram weights* of Table 4 show that this approach does not lead to an improvement.

Another possibility to compensate for the effect of unwanted backoff model prominence is eliminating small corpora RFs from the final interpolation. We thus eliminated 4 of the 48 corpora. The weights of the top corpora among the remaining 44 become less peaky but the results shown in the *FRF without small corpora* row in Table 4 give the impression this does not solve the problem.

We also tried to hand-edit the modified Kneser-Ney discount coefficients for order 4 making them equal to 0.1 and then re-estimate the individual RF models. This way we rely more on probability estimates for 4-grams provided by the RF models and pass less probability mass to the backoff N-gram LM. The results for such models are shown in the last two rows of Table 4.

The lattices generated by the Mandarin GALE Phase 4 LIMSI STT system were rescored with the best RFLM (the first one from the Table 4). The LM

Table 5. CER of RF on dev09s set

RF weight	0.00	0.10	0.15	0.20	0.25	0.30	0.50	1.00
CER	9.81	9.77	9.76	**9.72**	9.75	9.75	9.80	10.41

to generate these lattices is the baseline 4-gram LM described earlier. As we already mentioned, the lattices were not rescored with the neural network LM. These results are presented in Table 5. The *RF weight* row corresponds to the weights given to the RFLM. Small but significant improvement in CER over the baseline N-gram model is observed with the RFLM.

5 Conclusion and Future Work

Improving over a robust state-of-the-art STT system trained on large amounts of data is a very challenging task. Many of the approaches that perform well on small and medium-size tasks do not scale well to experiments on large data. In this paper we presented results using random forest language models to improve upon a well-tuned, competitive speech-to-text system for Mandarin Chinese. Improvements both in perplexity and CER were observed. However, these improvements are significantly less impressive than those reported for smaller scale tasks. This lessor degree of improvement can be expected for large scale tasks. One can argue that the RF approach can actually be regarded as a sophisticated smoothing technique. At the same time a baseline 4-gram LM with a comparatively small vocabulary of 56K words is trained on the very large corpus containing 3.2 billion words that makes the estimates provided by this model robust and rather reluctant to adding new ways of enhancements.

The results presented here are still preliminary. Due to the very large size of training data and high computational demands imposed by a random forest LM several simplifications were made at the stage of RF construction. E.g. the number of nodes was forced to be not much than 10000 (while in fully grown trees it can be more than a million), for very large corpora DT structures were not individually trained but only the corresponding counts were poured down the nodes of the trees trained on restricted data, etc. These simplifications may result in losing much of the potential gain that can be attained with RFLMs (for example in the forest of random forests scenario). Thus, the major direction of future work is performing efficient straightforward training of RF language models on the same amounts of data available for N-gram LM training.

Acknowledgments. This work has been partially supported by OSEO under the Quaero program and by the GALE program. Any opinions, findings or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

References

1. Bahl, L.R., Brown, P.F., de Souza, P.V., Mercer, R.L.: A Tree-Based Statistical Language Model for Natural Language Speech Recognition. CSL 37, 1001–1008 (1989)
2. Xu, P., Jelinek, F.: Random Forests in Language Modeling. In: Proc. of EMNLP 2004, Barcelona, pp. 325–332 (2004)

3. Navratil, J., Jin, Q., Andrews, W., Campbell, J.P.: Phonetic Speaker Recognition Using Maximum-Likelihood Binary Decision Tree Models. In: Proc. of ICASSP 2003, Hon Kong, pp. 796–799 (2003)
4. Xu, P.: Random Forests and the Data Sparseness Problem in Language Modeling. PhD Thesis, Johns Hopkins University, Baltimore (2005)
5. Su, Y., Jelinek, F., Khudanpur, S.: Large-Scale Random Forest Language Models for Speech Recognition. In: Proc. of Interspeech 2007, Antwerp, pp. 598–601 (2007)
6. Oparin, I.: Language Models for Automatic Speech Recognition of Inflectional Languages. PhD Thesis, University of West Bohemia, Plzen, Czech Republic (2009)
7. Oparin, I., Glembek, O., Burget, L., Černocký, J.: Morphological Random Forests for Language Modeling of Inflectional Languages. In: Proc. of IEEE Spoken Language Technology Workshop, SLT 2008, Goa, pp. 189–192 (2008)
8. Su, Y.: Knowledge Integration Into Language Models: A Random Forest Approach. PhD thesis, Johns Hopkins University, Baltimore (2009)
9. Luo, J., Lamel, L., Gauvain, J.-L.: Modeling Characters Versus Words for Mandarin Speech Recognition. In: Proc. of ICASSP 2009, Taipei, pp. 4325–4328 (2009)
10. Wu, D., Fung, P.: Improving Chinese Tokenization with Linguistic Filters on Statistical Lexical Acquisition. In: Proc. of ANLP 1994, pp. 180–181 (1994)
11. Sproat, R., Chilin, S., Gale, W., Chang, N.: A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. Computational Linguistics 22(3), 218–228 (1996)
12. Gauvain, J.L., Lamel, L., Adda, G.: The LIMSI Broadcast News Transcription System. Speech Communication 37(1-2), 89–108 (2002)
13. Lamel, L., Messaoudi, A., Gauvain, J.-L.: Improved Acoustic Modeling for Transcribing Arabic Broadcast Data. In: Proc. of Interspeech 2007, Antwerp, pp. 2077–2800 (2007)
14. Leggetter, C.J., Woodland, P.C.: Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. CSL 9(2), 171–185 (1995)
15. Hieronymus, J.L., Liu, X., Gales, M.J.F., Woodland, P.C.: Exploiting Chinese Character Models to Improve Speech Recognition Performance. In: Proc. of Interspeech 2010, Brighton, pp. 367–370 (2010)
16. Kneser, R., Ney, H.: Improved Backing-off for M-gram Language Modeling. In: Proc. of ICASSP 1995, Detroit, pp. 181–184 (1995)
17. Schwenk, H., Gauvain, J.-L.: Training Neural Network Language Models on Very Large Corpora. In: Proc. of EMNLP, Vancouver, pp. 201–208 (2005)
18. Su, Y.: Random Forest Language Model Toolkit, http://www.clsp.jhu.edu/~yisu/rflm.html

Design and Evaluation of an Agreement Error Detection System: Testing the Effect of Ambiguity, Parser and Corpus Type

Maite Oronoz, Arantza Díaz de Ilarraza, and Koldo Gojenola

IXA NLP Group
University of the Basque Country
{maite.oronoz,a.diazdeilarraza,koldo.gojenola}@ehu.es
http://ixa.si.ehu.es

Abstract. We present a system for the detection of agreement errors in Basque, a language with agglutinative morphology and free order of the main sentence constituents. Due to their complexity, agreement errors are one of the most frequent error types found in written texts. As the constituents concerning agreement can appear in any order in the sentence, we have implemented a system that makes use of dependency trees of the sentence, which abstract over specific constituent orders. We have used *Saroi*, a tool that obtains the analysis trees that fulfill a set of restrictions described by means of declarative rules. This tool is applied to the output of two dependency analyzers: *MaltIxa* (data-driven) and *EDGK* (rule-based). The system has been evaluated on two corpora: a group of texts containing errors, and another one composed of correct texts. As a secondary result, we have also estimated a measure of the impact of syntactic ambiguity on the quality of the results.

Keywords: Grammar error, ambiguity, parsing.

1 Introduction

Detection of grammatical errors is a relevant area of study in computer-assisted language learning and grammar checking. This paper presents the implementation and evaluation of a system for the detection of agreement errors in Basque, regarded as one of the most frequent kinds of error [1].

Referring to the process of detecting grammatical errors, Díaz de Ilarraza et al. [2] distinguish between *local* and *global* errors depending on the context they appear. Agreement errors belong to the second category due to the fact that the process of finding them is not limited to a local context (that is, a window of five or six consecutive words) but their detection requires the use of full sentence contexts, as the types of elements involving agreement (subject-verb, object-verb, indirect object-verb) may appear far from each other in the sentence. For this reason, our system will make use of syntactic dependency trees, which have the property of abstracting over specific constituent orders.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 281–292, 2010.

After the analysis of dependencies the system will make use of *Saroi* [3], a tool that, given a set of dependency trees, obtains those that fulfill the set of restrictions described by means of declarative rules. Although the tool is useful for several types of tree inspection processes, in this work we will use it for the detection of ungrammatical structures. *Saroi* will be applied to the outputs of two dependency analyzers: *EDGK*, a knowledge-based dependency parser [4]; and, *MaltIxa* [5] a data-driven parser based on Maltparser, a freely available and state of the art parser [6]. For the evaluation of the system, texts containing errors and correct texts from the Basque Dependency Treebank [7] will be used.

We are also concerned about the impact of morphosyntactic ambiguity in the quality of our system. A lot of error detection has been carried out on English, for which this kind of ambiguity is less of an issue, but in morphologically rich languages, a deep analysis of the influence of ambiguity in error detection is, in our opinion, fundamental. Among the three main types of ambiguity that can be relevant to grammatical error treatment (morphological, syntactic and semantic), our study will concentrate on measuring the effect of morphological and syntactic ambiguity in the results, leaving aside semantic ambiguity.

The remainder of this paper is organized as follows. After this introduction, section 2 relates our work to similar systems. Section 3 comments on general aspects of agreement errors in Basque. Section 4 will describe the linguistic resources used for the analysis of incorrect texts (corpora), and the main computational tools: two dependency analyzers and *Saroi*, a tool for tree inspection. Section 5 will present the experiments performed and the main results obtained. We conclude the paper in section 6 with our main contributions.

2 A Bird's Eye View of Error Detection Techniques

Approaches to grammatical error detection/correction are difficult to compare due to mainly the following reasons: i) most of them concentrate on one error type, and ii) the lack of large available error corpora. Choosing the more appropiate technique to the problem of error detection is not a trivial decision. Empirical and knowledge-based approaches can be used for this purpose.

Empirical approaches are suitable for error types related to the omission, replacement or addition of elements. For example, Tetreault and Chodorow [8] use machine learning techniques to detect errors involving prepositions in non-native English speakers. A deeply studied area using machine learning techniques is that of "context-sensitive spelling correction" [9], where the objective is to detect errors due to word confusion (e.g. *to/too*). Bigert and Knutsson [10] prove that precision is significantly improved when unsupervised methods are combined with linguistic information.

Regarding knowledge-based methods, many types of "local syntactic errors" have been detected by means of tools based on finite-state automata or transducers, such as Constraint Grammar (CG) [11], The Xerox Finite State Tool [12] or *ad hoc* systems. Systems based on finite state techniques usually define error patterns encoded in the form of rules which are applied to the analyzed texts.

For "global error" treatment, approaches based on context free grammars (CFG) or finite state techniques have been used. For example, CFG-based systems have experimented with the "relaxation" of some constraints in the grammar [13] or have specially developed error grammars [14]. Statistical parsers have also been used, that get a measure of grammaticality [15].

The relationship between ambiguity and error detection has been mentioned in very few occasions [16,17]. Similarly to most NLP areas, the development of tools for grammatical error detection finds ambiguity as a main obstacle for the design of efficient and accurate systems. Birn [16] states that the errors accumulated through morphological and syntactic analysis make it difficult to detect grammatical errors.

3 Agreement Errors in Basque

Basque is an agglutinative language with free order among the elements of the sentence. When classifying the errors related to agreement, we can distinguish three types of contexts:

- Intra-sentence agreement. The subject, object and indirect object must agree with the verb in case, number and person. These constituents can appear in any order in the sentence. It can appear in simple or compound sentences.
- Intra-phrase agreement. The constituents inside a phrase (e.g., a determiner) must agree with the head of the phrase (e.g., a noun).
- Other types of agreement. For example, an apposition and its corresponding main clause must agree in case, number and person.

We performed a manual study on the frequency of each type of error over a sample of 64 sentences containing agreement errors (and, sometimes, other types of error) that were taken from a database containing grammatical errors, and we found that intra-sentence agreement was by far the most common type (59 of the sentences, compared to 5 intra-phrase errors). For that reason, we dedicated our effort to this kind of error. Table 1 shows an example of a typical agreement error, where the verb must agree with the main grammatical elements (subject and object) in case, number and person. The fact that these elements can appear in any order with respect to the verb and also to each other makes error detection a difficult task, as there is a high number of possible permutations.

In brief, intra-sentence agreement errors can be abstracted as a local dependency tree where the main verb is the head, and the subject, object and indirect object are the dependents, together with the auxiliary that is also a dependent of the main verb and contains agreement information about the gramatical relations (case, number and definiteness).

Table 1. Agreement error (SBJ: subject, OBJ: object, ERG: ergative, ABS: absolutive)

**Zentral nuklear-r-ak*	*zakar erradiaktiboa*	*eratzen dute*
Power_station nuclear-0-the/ABS/PL/DET	rubish radioactive/ABS/SG	create AUX/SBJ:ERG_3PL,OBJ:ABS_3SG
'*The nuclear power station'	'create'	'radioactive rubbish'

4 General Linguistic Resources

4.1 The Corpus

The task of creating large sets of ungrammatical sentences is a necessary but time consuming activity. A corpus of this type can be composed of sentences produced by language learners (learner corpus) or it can be taken from a general error corpus not necessarily produced in a language-learning context [14]. Although some approaches propose the automatic creation of ungrammatical sentences [18], we decided to use a set of genuine errors. For evaluation, we use two corpora:

- A general purpose error corpus. It contains 1,000,000 words collected from different sources (language schools, technical reports, e-mails...). For the current experiments, we took a small subset of this corpus (5,000 words or 267 sentences), in which agreement errors were manually annotated.
- The Basque Dependency Treebank (BDT). This is a collection of presumably correct texts, that contains 55,000 tokens. Working with correct texts allows us to test the system negatively, that is, we test the system's behavior regarding false alarms, an important facet in automatic error detection.

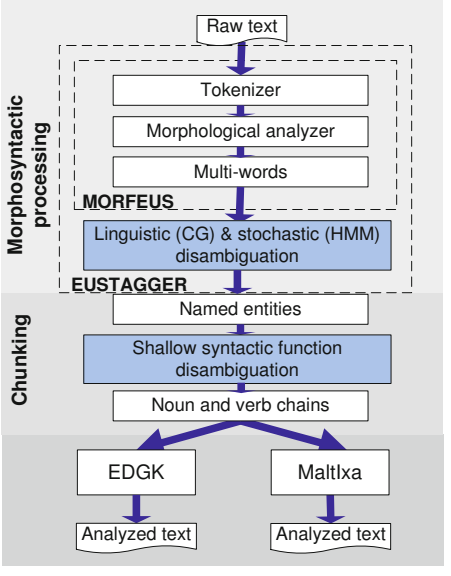
4.2 Syntactic Analysis

The creation of NLP tools is a very expensive task, so, instead of preparing specially tailored resources for error processing we decided to use the existing systems in our group, and perform the necessary adaptations to deal with ill-formed sentences. For the analysis of the input texts, we use the syntactic analysis chain for Basque [19]. It is composed of three main components (see figure 1):

- *Morphosyntactic processing.* It includes tokenization, morphological analysis, and detection of multiwords, followed by morphological disambiguation.
- *Chunking.* It detects named entities, and, after a shallow syntactic function disambiguation phase, obtains nominal and verbal chunks.
- *Dependency parsing.* A parser obtains dependency trees.

There are two modules in charge of disambiguation (see figure 1):

- *Morphosyntactic disambiguation* (linguistic and stochastic disambiguation). After applying the morphological analyzer (MORFEUS), the tagger/lemmatizer EUSTAGGER obtains the lemma and category of each form, also performing disambiguation using the part of speech (POS), fine grained POS (SubPOS) or case. Disambiguation is carried out by means of linguistic rules using Constraint Grammar (CG) and stochastic techniques [20]. Figure 1 shows the parameterizable disambiguation levels in EUSTAGGER. M1, M2 and M3 combine linguistic and stochastic disambiguation using different linguistic features, while M4 only uses CG. M3 is the option that disambiguates the most (95.42% precision).



Level	Linguistic features	Method
M1	POS	CG + HMM
M2	POS + SubPOS	CG + HMM
M3	POS + SubPOS + Case	CG + HMM
M4	M3 + rest of features	CG
S1	SF nom. & verb. chunks	CG
S2	S1 + main SF	CG

Fig. 1. The syntactic analysis chain for Basque and the disambiguation levels in it

– *Shallow syntactic function disambiguation.* This is carried out in two levels, which disambiguate different kinds of functions (see figure 1):
  - S1: it deals with syntactic functions (SF) related to nominal and verbal chunks. That is, functions internal to chunks.
  - S2 treats all functions in S1 plus the main syntactic functions.

Two dependency-based parsers have been used in the present work:

– *EDGK*, a knowledge-based dependency parser [4] based on CG.
– *MaltIxa*, an adaptation of Maltparser, a data-driven dependency parser [6] successfully applied to typologically different languages and treebanks.

Regarding accuracy, *EDGK* obtains 48% precision and 46% recall on well-formed texts while *MaltIxa* obtains 76.76% LAS[1]. Although the results are not directly comparable, they serve as an estimate of each parser's performance.

4.3 Saroi: A Tool for Inspecting Dependency Trees

For the detection of agreement errors we applied *Saroi*, a system developed to apply a set of query-rules to dependency trees. *Saroi* takes as input a group of analysis trees and a group of rules, and obtains as output the dependency trees that fulfill the conditions described in the rules. Its main general objective is the analysis of any linguistic phenomena in corpora.

[1] Labeled Attachment Score.

```
AGREEMENT_SUBJ_CASE_N_NK (
Detect (
  @!ncsubj!ncmod∼ &
  @!auxmod.type == 'nor-nork' &
  @!ncsubj!ncmod.case !=   )
```

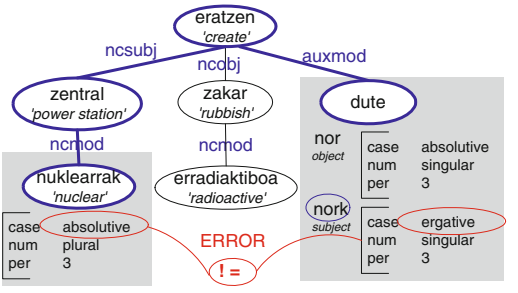


Fig. 2. A rule (left side) detecting the agreement error in the dependency tree (right side) of the sentence in Basque (*Nuclear power station create radioactive rubbish)

Figure 2 shows an example of a rule that detects the error in the dependency tree of the same figure. In the sentence the subject *zentral nuklearrak* (nuclear power station), in absolutive case, and the auxiliary verb, *dute* (linked to the main verb *eratzen*, create) and which needs a subject in ergative, do not agree.

Saroi uses as input the result of the syntactic analyzer (see section 4.2), in which the relations between the elements of the sentence are ambiguous, as a result of the remaining morphosyntactic ambiguity (see figure 3 in which, for example, "nuklearrak" has 3 interpretations). Then, *Saroi* constructs all the set of non ambiguous trees starting from an initially ambiguous tree (figure 3). The detection rules are applied to the expanded set of dependency trees.

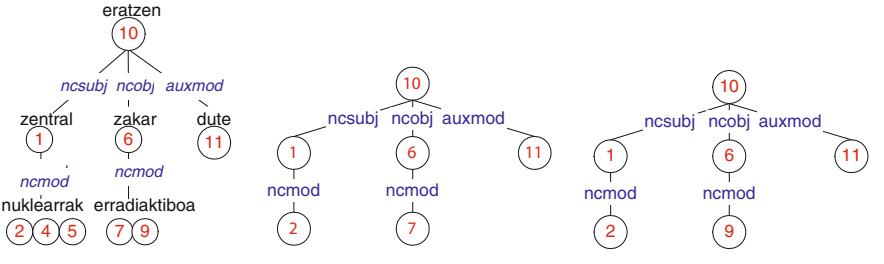


Fig. 3. Ambiguous tree and some of its corresponding non ambiguous trees

5 Experiments

In this section we will first comment on the experimental settings of the evaluation (5.1 and 5.2), and then we will present the results obtained (5.4).

5.1 Preprocessing

When using the predefined linguistic analysis chain for the detection of agreement errors, we had to take several aspects into account:

– *Difficulties due to language features (ellipsis and ambiguity).* In Basque "The phrases that agree with the verb need not be overtly manifest in the sentence: ergative, dative and absolutive noun phrases or pronouns can be absent and understood[2]". The inherent ambiguity of ellipsis together with semantic ambiguity make it difficult to decide on the correctness of a sentence.
– *Difficulties inherent to automatic language processing.* One of the syntactic analyzers (EDGK) obtains *partial analyses*, that is, not all the elements of the sentence appear in the final dependency trees, due to lack of coverage of the parser. Additionally, the errors mount up in the analysis chain, increasing the number of false alarms.

In an effort to overcome the mentioned problems, we decided to add a preprocessing module that will enrich dependency trees in three ways:

– Enriching nodes corresponding to coordination. For example, when two singular NPs are coordinated, the resulting constituent will agree in plural. For this task we used a set of CG rules.
– Enriching the auxiliary verb with agreement information about case, number and person of the subject, object and indirect object, that was not explicitly shown but was implicitly known. For example, the auxiliary verb *dute* indicates that the subject is *haiek* ('those') and the object *hura* ('that'). We made explicit, for example, that the subject has the features: case= "ergative", number="plural" and person="3".
– Enriching verbs with subcategorization information relevant for agreement, using patterns extracted from three data sources: i) manually developed schemas ii) realization-schemas automatically extracted from a corpus and, iii) information about auxiliary verbs from a dictionary.

5.2 Evaluation Methodology

Considering the problems mentioned in section 5.1 and being concerned about the impact of ambiguity in the quality of our analyzers, we followed these steps:

1. We chose the best option for morphological and syntactic disambiguation.
2. Once we decided the appropiate disambiguation level, we evaluated the system using two corpora: correct and error corpora.

An important remark regarding evaluation is that we will not apply the standard development-refinement-test cycle, but instead we will follow a development-test methodology: a) design of error detection rules in Saroi, and, b) evaluation. This means that there will not be a second step for the refinement of the rules after examining their results on a development set. Our aim was to test the effectiveness of a set of *clean* error detecting rules over different settings (corpus, parser and ambiguity). In that respect, the rules examine *clear* (and possibly naive) linguistic statements (e.g. the subject and verb must agree in case and number). This also means that there will be room for improvement of the results, after adapting the error detection rules to the details of real and/or noisy data.

[2] http://www.ei.ehu.es/

5.3 Election of the Disambiguation Level

Due to morphosyntactic and syntactic ambiguity, a number of trees ranging from 1 to more than 100 are generated for each sentence. Taking into account the combinations of morphosyntactic and shallow syntactic function disambiguation levels, the best disambiguation criteria should be those that: a) detect the highest number of errors in ungrammatical sentences, b) give the lowest number of false alarms in grammatical sentences, and c) generate the lowest number of analysis trees for each sentence (efficiency). With this objective, we followed two steps:

1. First, we chose the best morphosyntactic disambiguation level.
2. Second, after the morphosyntactic disambiguation level was fixed, we selected the best option for shallow syntactic function disambiguation.

For that reason, we selected a set of 10 ungrammatical sentences and their respective corrections (one for each sentence, that is, a total of 20 sentences). The sentences were analyzed with the eight disambiguation combinations[3] giving the results shown in table 2. The two combinations that generate the lowest number of trees with acceptable detection and false alarm rates were those performing the deepest morphosyntactic disambiguation, that is, M3[4] (S1 and S2).

Table 2. Looking for the best morphosyntactic disambiguation-combination

Disambiguation combinations	M1-S1	M2-S1	**M3-S1**	M4-S1	M1-S2	M2-S2	**M3-S2**	M4-S2
Number of trees	67.7	67.7	**27.8**	46.7	22.11	22.11	**11.6**	11.62
Errors in ungrammatical	5	5	**6**	6	5	5	**6**	6
False alarms in grammatical	0	0	**1**	1	0	0	**1**	0

Next, we performed a deeper analysis to choose the best syntactic function disambiguation level (S1 or S2). We soon realized that the grammar that assigns the dependency relations to correct texts need of relaxation when applied to ill-formed ones. For example, in the sentence "**nik ez nago konforme*" (I do not agree), the word "*nik*" (I) was not tagged as *subject* as it carries the ergative case, and the auxiliary verb asks for a subject in absolutive (this is a constraint in the dependency grammar when assigning the *subject* tag). We experimented relaxing all the conditions referred to the type of auxiliary in the rules assigning *subject*, *object* and *indirect object* relations. This relaxation is not performed for error detection (this is done by means of error detection rules) but it is necessary for the assigning of dependency relations to ungrammatical sentences. Then, in a second experiment, we used a set of 75 sentences containing agreement errors together with their corrections. The sentences were analyzed with the M3-S1-Relaxed, M3-S1-NotRelaxed, M3-S2-Relaxed and M3-S2-NotRelaxed combinations. The best results were obtained with the M3-S2-Relaxed option,

[3] 8 combinations: 4 morphosyntactic * 2 syntactic.

[4] Although the M4-S2 combination in table 2 seems to be good, it sometimes creates too many trees and, in other cases, it does not obtain any analysis tree.

that is, the option that disambiguates the most and with the relaxed dependency relation assignment. A deeper study about the impact of ambiguity in error detection is described in [2].

5.4 Evaluation of the System

After these tests, we noticed that the results are directly proportional to the parser's accuracy. When the relations are wrongly assigned, the detection of agreement errors is difficult. Sometimes, a false detection occurs, that is, an erroneous sentence is flagged as incorrect, but with a rule that is not the expected one. The rules mark the sentence as incorrect, but they fail in the diagnosis.

Correct corpora. We evaluated our system against the Basque Dependency Treebank. Its relations are presumably perfect (there is no need of a parser, neither the problem of ambiguity nor partial parsing), so the system should perform well. This experiment served to evaluate the system on false alarms. A subset containing 1906 trees was used. After applying the detection rules, 161 errors were flagged (8.45 % of the corpus). As this implies a high false alarm rate, we made a detailed analysis (table 3), finding out that:

- 90 of the cases were due to incorrect tagging, and *could not be considered false alarms*. In 41 of these sentences the error rule was applied because of treebank tagging decisions associated to special phenomena (e.g., in cases of an elliptical verb, two subjects were attached to the same verb when this is not grammatically correct) while in 49, annotation errors were detected (e.g. the object and the subject were mixed up because in Basque sometimes they take the same form...). So they correspond to treebank tagging errors.
- In 63 of the cases a *false alarm* (FA) occurs. In the great majority of the cases (58), the FA was flagged due to the lack of information in the verb subcategorization schemas. In these cases the verb appears with an unusual auxiliary verb (with complete subcategorization information, these are likely to disappear). The rest of the false alarms are very specific cases.
- In 8 cases a real agreement error occurs in the treebank.

In short, this experiment, apart from detecting false alarms in the treebank, also served to detect annotation mistakes, and gave us a measure of the importance of having correct verb subcategorization schemas.

Table 3. Evaluation results on the Basque Treebank

Flagged by the system	Numb.	From FA	From treebank
Not considered FA	90	55.9 %	4.72 %
FA	63	39.13 %	**3.30 %**
Real errors	8	4.97 %	
Total	161		8.45 %

Error corpora. We also performed an evaluation of the system on error corpora, using both *EDGK* and *MaltIxa*. We applied the agreement detection rules to all the possible analysis trees of the sentences. We calculated four results:

1. Using a data-driven parser (*MaltIxa*, **M**).
2. The knowledge-based parser (*EDGK*, **E**).
3. *MaltIxa* **and** *EDGK* (**M & E**). An error will be marked if it is flagged in the dependency trees obtained by *MaltIxa* and *EDGK*.
4. *MaltIxa* **or** *EDGK* (**M | E**). If an error is flagged on the output of either of the syntactic analyzers, the sentence will be deemed erroneous.

Examining the results in table 4 we see that, when applying the full set of error detection rules, precision varies between 24.26% and 26.19%. As could be expected, the best precision results were reached with the option **M & E** (when the error is flagged in the trees analyzed by both analyzers, the system is certain about the error). However, recall falls down (24.44%). In general, looking to both precision and recall, the two best options seem to be **M** and **M | E**. In general, the data-driven parser gets better results with correct texts, and it also behaves better with incorrect sentences, showing a robust behaviour.

There are two error detection rules (named TWO_SUBJ and TWO_OBJ) that account for most of the false alarms, both with EDGK and *MaltIxa*. These rules mark the attachment of two subjects (objects) to a verb. This phenomenon can occur as a consequence of a genuine agreement error, but also because of an incorrect dependency analysis, and is the reason for many false alarms. We think that as the frequency of incorrect analysis trees is relatively high, these rules cause more harm than good. For that reason, we perform three experiments to confirm this assumption. In the second row of table 4 we show the results without considering the rule that detects two subjects (TWO_SUBJ). In the third one, the rule that checks the appearance of two objects is removed (TWO_OBJ)

Table 4. Agreement error detection with *MaltIxa* and EDGK

		Correctly detected	FA	Detected	P	R	F
All the rules	**M**	28	81	109	**25.68 %**	**62.22 %**	36.35
	E	17	53	70	24.28 %	37.77 %	29.56
	M & E	11	31	42	**26.19 %**	24.44 %	25.28
	M | E	33	103	136	24.26 %	**73.33 %**	36.45
Without the rule TWO_SUBJ	**M**	26	59	85	30.58 %	57.77 %	39.99
	E	14	35	49	28.57 %	31.11 %	29.78
	M & E	10	21	31	32.25 %	22.22 %	26.31
	M | E	29	73	102	28.43 %	64.44 %	39.45
Without the rule TWO_OBJ	**M**	22	55	77	28.57 %	48.88 %	36.06
	E	12	39	51	23.52 %	26.66 %	24.99
	M & E	8	19	27	29.62 %	17.77 %	22.21
	M | E	26	75	101	25.74 %	57.77 %	35.61
Without the rules TWO_SUBJ **and** TWO_OBJ	**M**	21	33	54	**38.88 %**	**46.66 %**	42.41
	E	10	19	29	34.48 %	22.22 %	27.02
	M & E	8	10	18	**44.44 %**	17.77 %	25.38
	M | E	23	42	65	**35.38 %**	**51.11 %**	41.81
Number of errors							45
Number of words							4995

and, finally, the last row shows the result of removing both rules. The best results are obtained in the last case: 44.44% precision in the **M & E** option against the worse recall (17.77%, and f-score of 25.38). Considering precision and recall, *MaltIxa* gives the best results (38.88% precision and 46.66% recall, f-score 42.41).

6 Conclusions and Future Work

In this work we have presented a set of experiments on agreement error detection applied to an agglutinative and free constituent-order language. For this, we have used *Saroi*, a tool built for the inspection of dependency trees. The tool allows us to design restrictions by means of query-rules to be applied on the output of dependency parsers. In the evaluation we have experimented tuning the ambiguity of the analysis chain, we have used two general-purpose dependency parsers and two types of corpora.

When analyzing the Basque Dependency Treebank, we have detected ill-formed dependency trees, that is, manual annotation mistakes. Additionally, we have evaluated our system regarding false alarms, obtaining a false alarm rate of 3.30 %. Most of the alarms could be easily avoided improving the verb subcategorization schemas we use, and in this way leaving a minimal false alarm rate. In consequence, we think that the precision of our grammar rules is high.

One of the main problems is the lack of coverage of the dependency analyzers. When the trees are not syntactically well-formed, the system is more prone to signal a false alarm. Any improvement in syntactic analysis will have a positive effect on the error detection system. In the future, we want to analyze how complementary are MaltIxa and EDGK, and how they could be combined to obtain suitable analysis trees.

Working with real texts also led us to consider the problem of ambiguity. The best results are obtained when using the deepest disambiguation level (both morphosyntactic and syntactic). This can be explained by the explosion in the number of trees when "all" the ambiguity is considered.

References

1. Zubiri, I.: Gramática didáctica del euskara. Didaktiker, Bilbo (1994)
2. Díaz de Ilarraza, A., Gojenola, K., Oronoz, M.: 2009. Evaluating the Impact of Morphosyntactic Ambiguity in Grammatical Error Detection. In: Recent Advances in Natural Language Processing 2009, Borovets, Bulgary (2009)
3. Díaz de Ilarraza, A., Gojenola, K., Oronoz, M.: 2005. Design and development of a system for the detection of agreement errors in Basque. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 793–802. Springer, Heidelberg (2005)
4. Aranzabe, M.: Dependentzia-ereduan oinarritutako baliabide sintaktikoak: zuhaitzbankua eta gramatika konputazionala. Ph.D. thesis, UPV-EHU (2008)
5. Bengoetxea, K., Gojenola, K.: Exploring treebank transformations in dependency parsing. In: RANLP 2009, Borovets, Bulgaria (2009)
6. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryiğit, G., Kübler, S., Marinov, S., Marsi, E.: Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(2), 95–135 (2007)

7. Aduriz, I., Aranzabe, M., Arriola, J.M., Atutxa, A., Díaz de Ilarraza, A., Garmendia, A., Oronoz, M.: Construction of a Basque dependency treebank. In: TLT 2003. Second Workshop on Treebanks and Linguistic Theories, Vaxjo, Sweden (2003)
8. Tetreault, J.R., Chodorow, M.: The ups and downs of preposition error detection in esl writing. In: Proceedings of Coling, Manchester (2008)
9. Carlson, A.J., Rosen, J., Roth, D.: Scaling up context-sensitive text correction. In: Proceedings of the Thirteenth Innovative Applications of Artificial Intelligence Conference (IAAI 2001), Menlo Park CA (2001)
10. Bigert, J., Knutsson, O.: Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In: Romand 2002, Italy (2002)
11. Karlsson, F., Voutilainen, A., Heikkila, J., Anttila, A.: Constraint Grammar: Language-independent System for Parsing Unrestricted Text, Berlin (1995)
12. Karttunen, L., Gaál, T., Kempe, A.: Xerox finite state tool. Technical report, Xerox Research Centre Europe (1997)
13. Teixeira Martins, R., Hasegawa, R., Volpe Nunes, M., Montilha, G., De Oliveira Jr., O.N.: Linguistic issues in the development of ReGra: A grammar checker for Brazilian Portuguese. NLE 4(4), 287–307 (1998)
14. Foster, J., Vogel, C.: Good reasons for noting bad grammar: Constructing a corpus of ungrammatical language. In: International Conference on Linguistic Evidence: Empirical, Theoretical and Computational Perspectives, Tübingen, Germany (2004)
15. Wagner, J., Foster, J.: The effect of correcting grammatical errors on parse probabilities. In: Proceedings of the 11th International Conference on Parsing Technologies (IWPT 2009), Paris, France (2009)
16. Birn, J.: Detecting grammar errors with Lingsoft's Swedish grammar-checker. In: Proceedings from the 12th Nordiske datalingvistikkdager, Nordgard (2000)
17. Hashemi, S.H.: Detecting Grammar Errors in Children's Writing: A Finite State Approach. In: Proceedings of the 13th Nordic Conference on Computational Linguistics (NoDaLiDa 2001), Uppsala, Sweden (2000)
18. Foster, J., Andersen, O.E.: Generrate: Generating errors for use in grammatical error detection. In: Proceedings from the 4th Workshop on Innovative Use of NLP for Building Educational Applications (2009)
19. Aduriz, I., Aranzabe, M., Arriola, J.M., Díaz de Ilarraza, A., Gojenola, K., Oronoz, M., Uria, L.: A cascaded syntactic analyser for Basque. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 124–134. Springer, Heidelberg (2004)
20. Ezeiza, N., Aduriz, I., Alegria, I., Arriola, J.M., Urizar, R.: Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In: COLING 1998, Montreal (1998)
21. Gojenola, K., Sarasola, K.: Aplicación de la relajación gradual de restricciones para la detección y corrección de errores sintácticos. In: Actas de SEPLN 1994, Córdoba, Spain (1994)
22. Golding, A.R., Roth, D.: A Winnow-Based Approach to Context-Sensitive Spelling Correction. Machine Learning 34(1-3), 107–130 (1999)

TectoMT: Modular NLP Framework

Martin Popel and Zdeněk Žabokrtský

Charles University in Prague
Institute of Formal and Applied Linguistics
{popel,zabokrtsky}@ufal.mff.cuni.cz

Abstract. In the present paper we describe TectoMT, a multi-purpose open-source NLP framework. It allows for fast and efficient development of NLP applications by exploiting a wide range of software modules already integrated in TectoMT, such as tools for sentence segmentation, tokenization, morphological analysis, POS tagging, shallow and deep syntax parsing, named entity recognition, anaphora resolution, tree-to-tree translation, natural language generation, word-level alignment of parallel corpora, and other tasks. One of the most complex applications of TectoMT is the English-Czech machine translation system with transfer on deep syntactic (tectogrammatical) layer. Several modules are available also for other languages (German, Russian, Arabic). Where possible, modules are implemented in a language-independent way, so they can be reused in many applications.

Keywords: NLP framework, linguistic processing pipeline, TectoMT.

1 Introduction

Most non-trivial NLP (natural language processing) applications exploit several tools (e.g. tokenizers, taggers, parsers) that process data in a pipeline. For developers of NLP applications it is beneficial to reuse available existing tools and integrate them in the processing pipeline. However, it is often the case that the developer has to spend more time with the integration and other auxiliary work than with the development of new tools and innovative approaches. The auxiliary work involves studying documentation of the reused tools, compiling and adjusting the tools in order to run them on the developer's computer, training models if these are needed and not included with the tools, writing scripts for data conversions (the tools may require different input format or encoding), resolving incompatibilities between the tools (e.g. different tagsets assumed), etc. Such a work is inefficient and frustrating. Moreover, if it is done in an ad-hoc style, it must be done again for other applications.

The described drawbacks can be reduced or eliminated by using an NLP framework that integrates the needed tools, so the tools can be combined into various pipelines serving for different purposes. Most of the auxiliary work is already implemented in the framework and developers can focus on the more creative part of their tasks. Some frameworks enable easy addition of third-party tools (usually using so-called wrappers) and development of new modules within the framework.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 293–304, 2010.

In this paper, we report on a novel NLP framework called TectoMT.[1] In Sect. 2, we describe its architecture and main concepts. Sect. 3 concerns implementation issues. Finally, in Sect. 4, we briefly describe and compare other NLP frameworks.

2 TectoMT Architecture

2.1 Blocks and Scenarios

TectoMT framework emphasizes modularity and reusability at various levels. Following the fundamental assumption that every non-trivial NLP task can be decomposed into a sequence of subsequent steps, these steps are implemented as reusable components called *blocks*. Each block has a well defined (and documented) input and output specification and also a linguistically interpretable functionality in most cases. This facilitates rapid development of new applications by simply listing the names of existing blocks to be applied to the data. Moreover, blocks in this sequence (which is called *scenario*) can be easily substituted with an alternative solution (other blocks), which attempts at solving the same subtask using a different approach or method.[2]

For example, the task of morphological and shallow-syntax analysis (and disambiguation) for English text consists of five steps: sentence segmentation, tokenization, part-of-speech tagging, lemmatization and parsing. In TectoMT we can arrange various scenarios to solve this task, for example:

Scenario A	Scenario B
`Sentence_segmentation_simple`	`Each_line_as_sentence`
`Penn_style_tokenization` `TagMxPost`	`Tokenize_and_tag`
`Lemmatize_mtree`	`Lemmatize_mtree`
`McD_parser`	`Malt_parser`

In the scenario A, tokenization and tagging is done separately in two blocks (`Penn_style_tokenization` and `TagMxPost`, respectively), whereas in the scenario B, the same two steps are done in one block at once (`Tokenize_and_tag`). Also different parsers are used.[3]

[1] `http://ufal.mff.cuni.cz/tectomt/`

[2] Scenarios can be adjusted also by specifying parameters for individual blocks. Using parameters, we can define, for instance, which model should be used for parsing.

[3] `Penn_style_tokenization` is a rule-based block for tokenization according to Penn Treebank guidelines (`http://www.cis.upenn.edu/~treebank/tokenization.html`). `TagMxPost` uses Adwait Ratnaparkhi's tagger [1]. `Tokenize_and_tag` uses Aaron Coburn's `Lingua::EN::Tagger` CPAN module. `Lemmatize_mtree` is a block for English lemmatization handling verbs, noun plurals, comparatives, superlatives and negative prefixes. It uses a set of rules (about one hundred regular expressions inspired by `morpha` [2]) and a list of words with irregular lemmatization. `McD_parser` uses MST parser 0.4.3b [3], `Malt_parser` uses Malt parser 1.3.1 [4].

TectoMT currently includes over 400 blocks – approximately 140 blocks are specific for English, 120 for Czech, 60 for English-to-Czech transfer, 30 for other languages and 50 blocks are language-independent. Some of them contain only few lines of code, some solve complex linguistic phenomena. In order to prevent code duplications, many tools and routines are implemented as separate modules, which can be used in more blocks. TectoMT integrates NLP tools such as:

- five taggers for English, three taggers for Czech, one tagger for German, Russian and Spanish,
- two constituency parsers for English, two dependency parsers for English, three dependency parsers for Czech, two dependency parsers for German,
- a named entity recognizer for English, and two named entity recognizers for Czech.

New components are still being added, as there are more than ten programmers contributing to the TectoMT repository at present.

2.2 Applications

Applications in TectoMT correspond to end-to-end NLP tasks, be they real end-user applications (such as machine translation), or only NLP-related experiments. Applications usually consist of three phases:

1. conversion of the input data to the TectoMT internal format, possibly split into more files,
2. applying a scenario (i.e. a sequence of blocks) to the files,
3. conversion of the resulting files to the desired output format.

Technically, applications are often implemented as Makefiles, which only glue the three phases.

Besides developing the English-Czech translation system [5], TectoMT was also used in applications such as:

- machine translation based on Synchronous Tree Substitution Grammars and factored translation [6],
- aligning tectogrammatical structures of parallel Czech and English sentences [7],
- building a large, automatically annotated parallel English-Czech treebank CzEng 0.9 [8],
- compiling a probabilistic English-Czech translation dictionary [9],
- evaluating metrics for measuring translation quality [10],
- complex pre-annotation of English tectogrammatical trees within the Prague Czech English Dependency Treebank project [11],
- tagging the Czech data set for the CoNLL Shared Task [12],
- gaining syntax-based features for prosody prediction [13],
- experiments on information retrieval [14],
- experiments on named entity recognition [15],
- conversion between different deep-syntactic representations of Russian sentences [16].

2.3 Layers of Language Description

TectoMT profits from the stratificational approach to the language, namely it defines four layers of language description (listed in the order of increasing level of abstraction): raw text (word layer, w-layer), morphological layer (m-layer), shallow-syntax layer (analytical layer, a-layer), and deep-syntax layer (layer of linguistic meaning, tectogrammatical layer, t-layer).

The strategy is adopted from the Functional Generative Description theory [17], which has been further elaborated and implemented in the Prague Dependency Treebank (PDT) [18]. We give here only a very brief summary of the key points.

- **morphological layer (m-layer)**
  Each sentence is tokenized and each token is annotated with a lemma and morphological tag. For details see [19].
- **analytical layer (a-layer)**
  Each sentence is represented as a shallow-syntax dependency tree (a-tree). There is one-to-one correspondence between m-layer tokens and a-layer nodes (a-nodes). Each a-node is annotated with the so-called *analytical function*, which represents the type of dependency relation to its parent (i.e. its governing node). For details see [20].
- **tectogrammatical layer (t-layer)**
  Each sentence is represented as a deep-syntax dependency tree (t-tree). Autosemantic (meaningful) words are represented as t-layer nodes (t-nodes). Information conveyed by functional words (such as auxiliary verbs, prepositions and subordinating conjunctions) is represented by attributes of t-nodes. Most important attributes of t-nodes are: tectogrammatical lemma, functor (which represents the semantic value of syntactic dependency relation) and a set of grammatemes (e.g. tense, number, verb modality, deontic modality, negation).

  Edges in t-trees represent linguistic dependencies except for several special cases, most notable of which are paratactic structures (coordinations). In these cases, there is a difference between the *topological parent* of a node (i.e. the parent as it is saved in the tree) and the *effective parent* (i.e. the governing node in a linguistic sense). Analogously, there is a notion of *topological children* and *effective children*. For details see [21].

Apart from the described layers, TectoMT also defines phrase-structure layer (p-layer) for storing constituency trees. This layer is approximately on the same level of abstraction as the a-layer.

2.4 Documents, Bundles and Trees

Every document is saved in one file and consists of a sequence of sentences. Each sentence is represented by a structure called *bundle*, which stands for 'a bundle of trees'. Each tree can be classified according to:

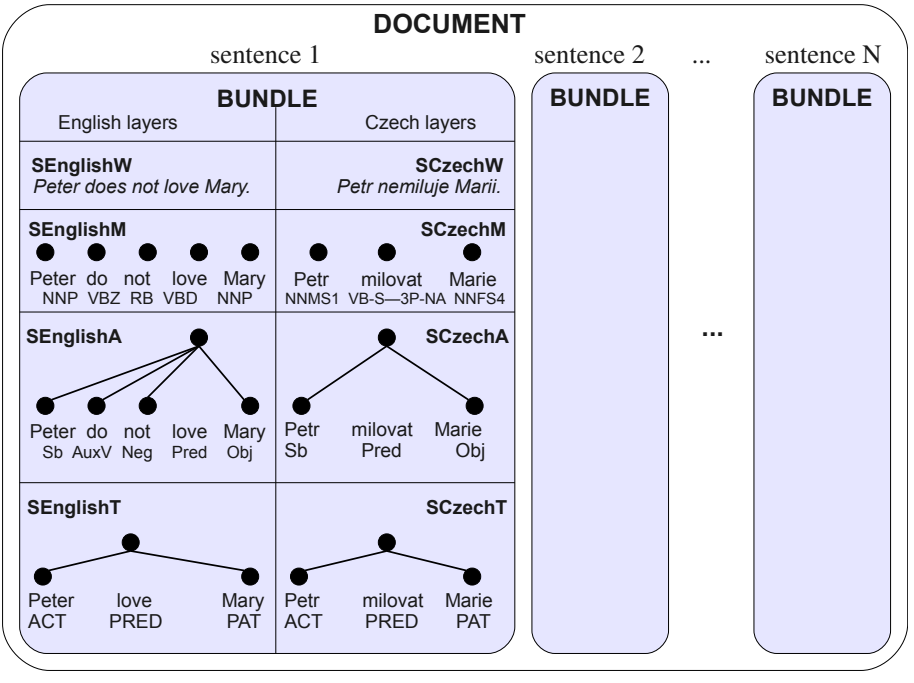


Fig. 1. English-Czech parallel text annotated on three layers of language description is saved in a TectoMT document, each sentence in one bundle. We show only a simplified representation of the trees in the first bundle.

- layer of language description (M=m-layer, A=a-layer, T=t-layer),
- language (e.g. Arabic, Czech, English, German[4]),
- indication whether the sentence was created by analysis (S=source) or by transfer or synthesis (T=target).

In other words, each bundle contains trees that represent the same sentence in different languages, layers and source/target direction (hence sentences in multilingual documents are implicitly aligned, see Fig. 1). TectoMT trees are denoted by the three coordinates, e.g. analytical layer representation of an English sentence acquired by analysis is denoted as `SEnglishA`, tectogrammatical layer representation of a sentence translated to Czech is denoted as `TCzechT`. This naming convention is used on many places in TectoMT: for naming blocks (see Sect. 3), for naming node identifiers, etc. The convention is extremely useful for a machine translation that follows the analysis-transfer-synthesis scheme as illustrated in Fig. 2 using Vauquois diagram. Nevertheless, also the blocks for other NLP tasks can be classified according to the languages and layers on which they operate.

[4] In the near future, TectoMT will migrate to using ISO 639 language codes (e.g. ar, cs, en, de) instead of full names.

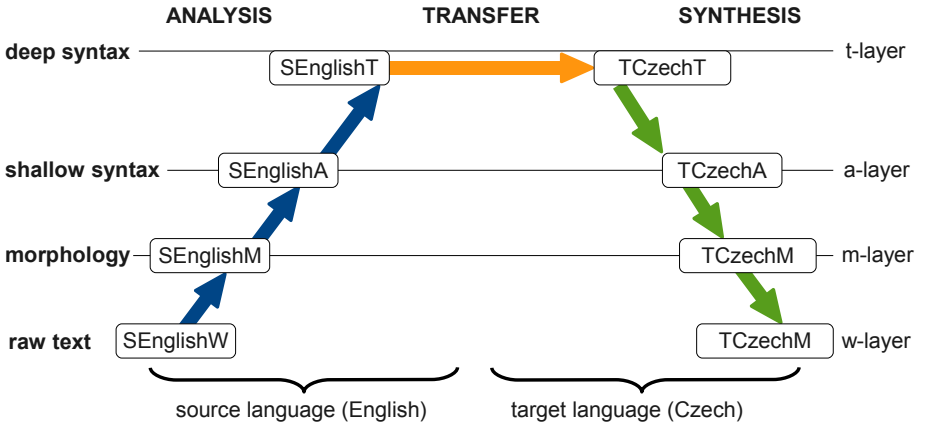


Fig. 2. Vauquois diagram for translation with transfer on tectogrammatical layer

A-layer and t-layer structures are dependency trees, so it is natural to handle them as tree data structures. M-layer structure is a sequence of tokens with associated attributes, which is handled in TectoMT as a special case of a tree (all nodes except the technical root are leaves of the tree). W-layer (raw text) is represented as string attributes stored within bundles.

3 TectoMT Implementation

3.1 Design Decisions

TectoMT is implemented in Perl programming language under Linux. This does not exclude the possibility of releasing platform-independent applications made of selected components (platform-independent solutions are always preferred in TectoMT). TectoMT modules are programmed in object-oriented programming style using inside-out classes (following [22]). Some of the modules are just Perl wrappers for tools written in other languages (especially Java and C).

TectoMT is a modern multilingual framework and it uses open standards such as Unicode and XML. TectoMT is neutral with respect to the methodology employed in the individual blocks: fully stochastic, hybrid, or fully rule-based approaches can be used.

3.2 TectoMT Components and Directory Structure

Each block is a Perl class inherited from `TectoMT::Block` and each block is saved in one file. The blocks are distributed into directories according to the languages and layers on which they operate. For example, all blocks for deep-syntactic analysis of English (i.e. for generating t-trees from a-trees) are stored in a directory `SEnglishA_to_SEnglishT`. In this paper, we use for simplicity only short names

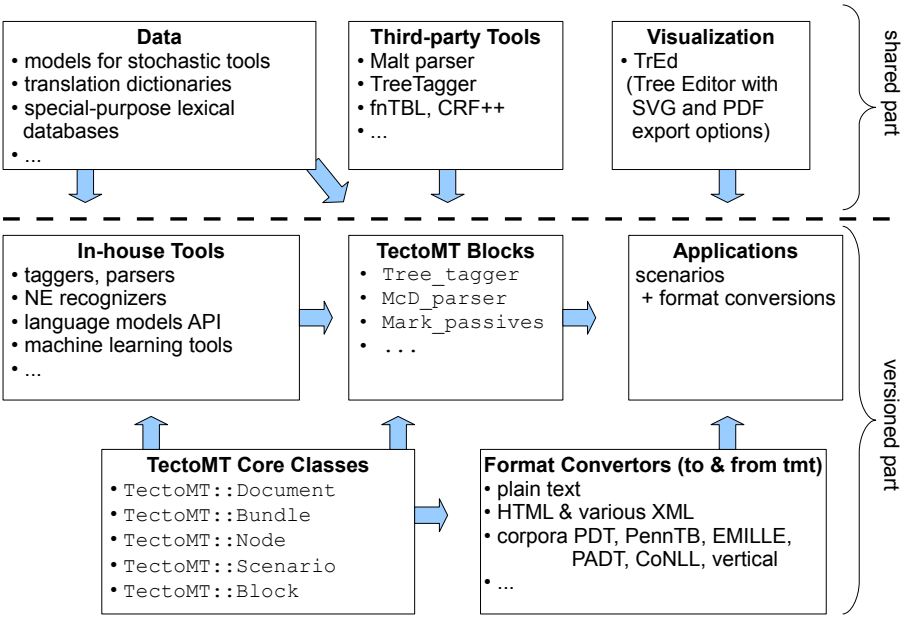


Fig. 3. Components of the TectoMT framework

of blocks, so e.g. instead of the full name `SEnglishA_to_SEnglishT::Assign_grammatemes` we write only `Assign_grammatemes`.

TectoMT is composed of two parts (see Fig. 3). The first part (the *versioned* part), which contains TectoMT core classes and utilities, format converters, blocks, applications, and in-house tools, is stored in an SVN repository, so that it can be developed in parallel by more developers. The second part (the *shared* part), which contains linguistic data resources, downloaded third-party tools and the software for visualization of TectoMT files (Tree editor TrEd [23]), is shared without versioning because (a) it is supposed to be changed rather additively, (b) it is huge, as it contains large data resources, and (c) it should be automatically reconstructible simply by downloading (and installing) the needed components.

3.3 Data Formats

The internal TectoMT format (`tmt` files) is an XML with a schema defined in Prague Markup Language [24]. It is similar to the format used for the Prague Dependency Treebank 2.0 [18], but all representations of a textual document at the individual layers of language description are stored in a single file. TectoMT includes converters for various formats and corpora (e.g. Penn Treebank [25], CoNLL [12], EMILLE [26], PADT [27]).

Scenarios are saved in plain text files with a simple format that enables including of other scenarios.

3.4 Parallel Processing of Large Data

TectoMT can be used for processing huge data resources using a cluster of computers.[5] There are utilities that take care of input data distribution, filtering, parallel processing, logging, error checks, and output data collection. In order to allow efficient and flawless processing, input data (i.e. corpora) should be distributed into many `tmt` files (with about 50 to 200 sentences per file). Each file can be processed independently, so this method scales well to any number of computers in a cluster. For example, the best version of translation from English to Czech takes about 1.2 seconds per sentence plus 90 seconds for initial loading of blocks in memory per computer (more precisely per cluster job).[6] Using 20 computers in a cluster we can translate 2000 sentences in less than 4 minutes.

TectoMT was used to automatically annotate the parallel treebank CzEng 0.9 [8] with 8 million sentences, 93 million English and 82 million Czech words.

4 Other NLP Frameworks

In Tab. 1, we summarize some properties of TectoMT and four other NLP frameworks:

- ETAP-3 is an NLP framework for English-Russian and Russian-English translation developed in the Computational linguistics laboratory of the Institute for Information Transmission Problems of the Russian Academy of Sciences. [28]
- GATE is one of the most widely used NLP frameworks with integrated graphical user interface. It is being developed at University of Sheffield. [29]
- OpenNLP[7] is an organizational center for open source NLP projects, which offers several NLP tools (and maximum entropy language models).
- WebLicht[8] is a Service Oriented Architecture (SOA) for building annotated German text corpora.

Each of the frameworks was developed for different purposes with different preferences in mind, so it is not possible to choose the universally best framework. There is a number of NLP frameworks for shallow analysis or translation (apart from those listed in Tab. 1, e.g. Apertium [31] or UIMA [32]). However, we are aware only of two tree-oriented (rather than sequence-oriented or chunk-oriented) frameworks capable of deep syntax analysis (and translation) – TectoMT and ETAP-3. Unlike ETAP-3, TectoMT is publicly available and allows for combining statistical and rule-based approaches (whereas ETAP-3 is strictly rule-based).

[5] We use Sun Grid Engine, `http://gridengine.sunsource.net/`

[6] Most of the initialization time is spent with loading translation and language models (about 8 GiB). Other applications presented in this paper are not so resource-demanding, so they are loaded in a few seconds.

[7] `http://opennlp.sourceforge.net`

[8] `http://weblicht.sfs.uni-tuebingen.de/englisch/index.shtml`

Table 1. Comparison of NLP frameworks. Notes:
a: ETAP-3 is a closed-source project, only a small demo is available online – http://cl.iitp.ru/etap
b: WebLicht is designed to offer web services for five universities in Germany, but we have not found any service for public use.
c: Functional Generative Description theory [17]
d: Meaning-Text Theory [30]
e: The purpose of NLP frameworks is to serve for various applications. However, some applications may be considered characteristic for a given framework.
MT = Machine Translation, IE = information extraction.

	TectoMT	ETAP-3	GATE	OpenNLP	WebLicht
developed since	2005	1980s	1996	2003	2008
license for public use	GPL	no[a]	LGPL	LGPL	no[b]
main prog. language	Perl	C/C++	Java	Java	?
linguistic theory	FGD[c]	MTT[d]			
strictly rule-based	no	yes	no	no	no
main application[e]	MT	MT	IE		annotation
uses deep syntax	yes	yes	no	no	no

5 Summary

TectoMT is a multilingual NLP framework with a wide range of applications and integrated tools. Its main properties are:

- **emphasized efficient development, modular design and reusability**
  There are cleanly separated low-level core routines (for processing documents, handling dependency trees and data serialization) and blocks for processing linguistic task. The users can easily add new blocks, which are written in a full-fledged programming language. TectoMT also integrates third-party tools and software for viewing the processed data.
- **stratificational approach to the language**
  TectoMT uses four layers of language description, which are linguistically interpretable (though this does not mean that TectoMT is a strictly rule-based framework). On the shallow and deep syntactic layer, sentences are represented as dependency trees. Annotation conventions are adopted mainly from commonly used corpora (PennTB, PDT).
- **unified object-oriented interface for accessing data structures**
  TectoMT tries to minimize file-based communication between the blocks in processing pipelines. The unified object-oriented interface allows for processing large amounts of data with complex data structures.
- **comfortable development**
  The analysis of sentences can be examined and edited in the tree editor TrEd. TectoMT also offers tools for parallel processing, testing and debugging, standard structure of blocks and documentation.

Acknowledgments. This work was supported by the grants GAUK 116310, MSM0021620838, MŠMT ČR LC536, and FP7-ICT-2007-3-231720 (EuroMatrix Plus). We thank three anonymous reviewers for helpful comments.

References

1. Ratnaparkhi, A.: A maximum entropy part-of-speech tagger. In: Proceedings of the conference on Empirical Methods in Natural Language Processing, pp. 133–142 (1996)
2. Minnen, G., Carroll, J., Pearce, D.: Robust Applied Morphological Generation. In: Proceedings of the 1st International Natural Language Generation Conference, Israel, pp. 201–208 (2000)
3. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In: Proceedings of Human Langauge Technology Conference and Conference on Empirical Methods in Natural Language Processing (HTL/EMNLP), Vancouver, BC, Canada, pp. 523–530 (2005)
4. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(2), 95–135 (2007)
5. Bojar, O., Mareček, D., Novák, V., Popel, M., Ptáček, J., Rouš, J., Žabokrtský, Z.: English-Czech MT in 2008. In: Proceedings of the Fourth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Athens, Greece, pp. 125–129 (March 2009)
6. Bojar, O., Hajič, J.: Phrase-Based and Deep Syntactic English-to-Czech Statistical Machine Translation. In: ACL 2008 WMT: Proceedings of the Third Workshop on Statistical Machine Translation, Association for Computational Linguistics, Columbus, OH, USA, pp. 143–146 (2008)
7. Mareček, D., Žabokrtský, Z., Novák, V.: Automatic Alignment of Czech and English Deep Syntactic Dependency Trees. In: Hutchins, J., Hahn, W. (eds.) Proceedings of the Twelfth EAMT Conference, Hamburg, HITEC e.V, pp. 102–111 (2008)
8. Bojar, O., Žabokrtský, Z.: Building a Large Czech-English Automatic Parallel Treebank. Prague Bulletin of Mathematical Linguistics 92 (2009)
9. Rouš, J.: Probabilistic translation dictionary. Master's thesis, Faculty of Mathematics and Physics, Charles University in Prague (2009)
10. Kos, K., Bojar, O.: Evaluation of Machine Translation Metrics for Czech as the Target Language. Prague Bulletin of Mathematical Linguistics 92 (2009)
11. Hajič, J., Cinková, S., Čermáková, K., Mladová, L., Nedolužko, A., Petr, P., Semecký, J., Šindlerová, J., Toman, J., Tomšů, K., Korvas, M., Rysová, M., Veselovská, K., Žabokrtský, Z.: Prague English Dependency Treebank, Version 1.0 (January 2009)
12. Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M.A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., Zhang, Y.: The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In: Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009), Boulder, Colorado, USA, June 4-5 (2009)

13. Romportl, J.: Zvyšování přirozenosti strojově vytvářené řeči v oblasti suprasegmentálních zvukových jevů. PhD thesis, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic (2008)
14. Kravalová, J.: Využití syntaxe v metodách pro vyhledávání informací (using syntax in information retrieval). Master's thesis, Faculty of Mathematics and Physics, Charles University in Prague (2009)
15. 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), Association for Computational Linguistics, Suntec, Singapore, pp. 194–201 (2009)
16. Mareček, D., Kljueva, N.: Converting Russian Treebank SynTagRus into Praguian PDT Style. In: Proceedings of the RANLP 2009, International Conference on Recent Advances in Natural Language Processing, Borovets, Bulgaria (2009)
17. Sgall, P.: Generativní popis jazyka a česká deklinace. Academia, Prague (1967)
18. Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M.: Prague Dependency Treebank 2.0. Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia (2006)
19. Zeman, D., Hana, J., Hanová, H., Hajič, J., Hladká, B., Jeřábek, E.: A Manual for Morphological Annotation, 2nd edn., Technical Report 27, ÚFAL MFF UK, Prague, Czech Republic (2005)
20. Hajičová, E., Kirschner, Z., Sgall, P.: A Manual for Analytic Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL MFF UK, Prague, Czech Republic (1999)
21. Mikulová, M., Bémová, A., Hajič, J., Hajičová, E., Havelka, J., Kolářová, V., Kučová, L., Lopatková, M., Pajas, P., Panevová, J., Razímová, M., Sgall, P., Štěpánek, J., Urešová, Z., Veselá, K., Žabokrtský, Z.: Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep (2006)
22. Conway, D.: Perl Best Practices. O'Reilly Media, Inc., Sebastopol (2005)
23. Pajas, P., Štěpánek, J.: Recent advances in a feature-rich framework for treebank annotation. In: Scott, D., Uszkoreit, H. (eds.) The 22nd International Conference on Computational Linguistics - Proceedings of the Conference, The Coling 2008 Organizing Committee, Manchester, UK, vol. 2, pp. 673–680 (2008)
24. Pajas, P., Štěpánek, J.: XML-based representation of multi-layered annotation in the PDT 2.0. In: Hinrichs, R.E., Ide, N., Palmer, M., Pustejovsky, J. (eds.) Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006), Genova, Italy, pp. 40–47 (2006)
25. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313–330 (1994)
26. McEnery, A., Baker, P., Gaizauskas, R., Cunningham, H.: EMILLE: Building a corpus of South Asian languages. Vivek-Bombay 13(3), 22–28 (2000)
27. Smrž, O., Bielický, V., Kouřilová, I., Kráčmar, J., Hajič, J., Zemánek, P.: Prague Arabic Dependency Treebank: A Word on the Million Words. In: Proceedings of the Workshop on Arabic and Local Languages (LREC 2008), Marrakech, Morocco, pp. 16–23 (2008)
28. Boguslavsky, I., Iomdin, L., Sizov, V.: Multilinguality in ETAP-3: Reuse of Lexical Resources. In: Sérasset, G. (ed.) COLING 2004 Multilingual Linguistic Resources, Geneva, Switzerland, August 28, pp. 1–8 (2004)

29. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, July 07-12 (2002)
30. Mel'čuk, I.A.: Towards a functioning model of language. Mouton (1970)
31. Tyers, F.M., Sánchez-Martínez, F., Ortiz-Rojas, S., Forcada, M.L.: Free/open-source resources in the Apertium platform for machine translation research and development. Prague Bulletin of Mathematical Linguistics 93, 67–76 (2010)
32. Wilcock, G.: Linguistic Processing Pipelines: Problems and Solutions. In: Book of Abstracts GSCL Workshop: Linguistic Processing Pipelines (2009)

Using Information from the Target Language to Improve Crosslingual Text Classification

Gabriela Ramírez-de-la-Rosa[1], Manuel Montes-y-Gómez[1], Luis Villaseñor-Pineda[1], David Pinto-Avendaño[2], and Thamar Solorio[3]

[1] Laboratory of Language Technologies,
National Institute for Astrophysics, Optics and Electronics
{gabrielarr,mmontesg,villasen}@inaoep.mx
[2] Faculty of Computer Science, Autonomous University of Puebla
dpinto@cs.buap.mx
[3] Department of Computer and Information Sciences,
University of Alabama at Birmingham
solorio@uab.edu

Abstract. Crosslingual text classification consists of exploiting labeled documents in a source language to classify documents in a different target language. In addition to the evident translation problem, this task also faces some difficulties caused by the cultural discrepancies manifested in both languages by means of different topic distributions. Such discrepancies make the classifier unreliable for the categorization task. In order to tackle this problem we propose to improve the classification performance by using information embedded in the own target dataset. The central idea of the proposed approach is that similar documents must belong to the same category. Therefore, it classifies the documents by considering not only their own content but also information about the assigned category to other similar documents from the same target dataset. Experimental results using three different languages evidence the appropriateness of the proposed approach.

Keywords: Crosslingual text classification, prototype-based method, unlabeled documents, text classification.

1 Introduction

Text classification is the task of assigning documents into a set of predefined classes or topics [1]. The leading approach for this task considers the application of machine learning techniques such as Support Vector Machines and Naïve Bayes, which require large labeled data sets to construct accurate classifiers. Unfortunately, due to the high costs associated with data tagging, for many applications in several languages these datasets are extremely small or, what is worst, they are not available.

Several approaches have recently proposed to alleviate the problem of lacking labeled data; one example is the *crosslingual text classification* (CLTC), which

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 305–313, 2010.

consists in exploiting labeled documents in a source language to classify documents in a different target language. Because of the inherent language-barrier problem of this approach, most current CLTC methods have mainly addressed different translation issues. In particular, they have explored the translation from one language to another by means of machine translation approaches as well as by multilingual lexical resources such as dictionaries and ontologies [2,3].

Although the language barrier is an important problem in CLTC, it is not the only one. It is clear that, in spite of a perfect translation, there are also some *cultural discrepancies* manifested in both languages that will affect the classification performance. That is, given that a language is the way of expression of a cultural and socially homogeneous community, documents from the same category but different languages (i.e., different cultures) may concern very different topics. As an example, consider the case of news about sports from France (in French) and from USA (in English); while the first will include more documents about soccer, rugby and cricket, the latter will mainly consider notes about baseball, basketball and American football. In order to tackle this problem, recent CLTC methods have proposed to enhance the classification model by iteratively incorporating information from the target language into the training phase [4,5,6]; their purpose is to obtain a classification model that is as close as possible to the target topic distribution.

The method proposed in this paper is a simple and inexpensive alternative for facing the problems caused the cultural discrepancies between both languages. Different to previous iterative approaches, it does not consider the modification or enrichment of the original classifier; instead, it attempts to improve the document classification by using more information to support the decision process. Mainly, it is based on the idea that similar documents must belong to the same category and, therefore, it classifies the documents by considering their own information (as usual) as well as the information about the assigned category to other similar documents from the same target dataset.

In the following section we describe the proposed method for CLTC. This method is based on the prototype-based classification approach [7], but modifies the traditional class-assignment strategy in order to incorporate information from the set of similar documents. Then, in Section 3 we define the experimental configuration and show results in six different pairs of languages that demonstrate the usefulness of the proposed approach for CLTC. Finally, in Section 4 we present our conclusions and some ideas for future work.

2 Prototype-Based CLTC Method

Given that prototype-based classification is very simple and has demonstrated to consistently outperform other algorithms such as Naïve Bayes, K-Nearest Neighbors and C4.5 in text classification tasks [7], we decided to implement the proposed approach using this classification algorithm. In general, our *prototype-based CLTC method* chooses a category for each document (from the target language) by determining the class which prototype (calculated from the source-language

training set) is more similar to it and to its nearest neighbors (from the same target-language dataset).

Figure 1 shows the general schema of the proposed method. It consists of four main processes. The first one carries out the translation of the training documents from the source language (S) to the target language (T). The second process focuses on the construction of the class prototypes using the well-known normalized sum technique [8]. The third process involves the identification of the nearest neighbors for each document from the target language dataset (D_T). Finally, the fourth process computes the classification for each document $d \in D_T$ considering information from their own and their neighbors. Bellow we present a brief description of each one of these processes.

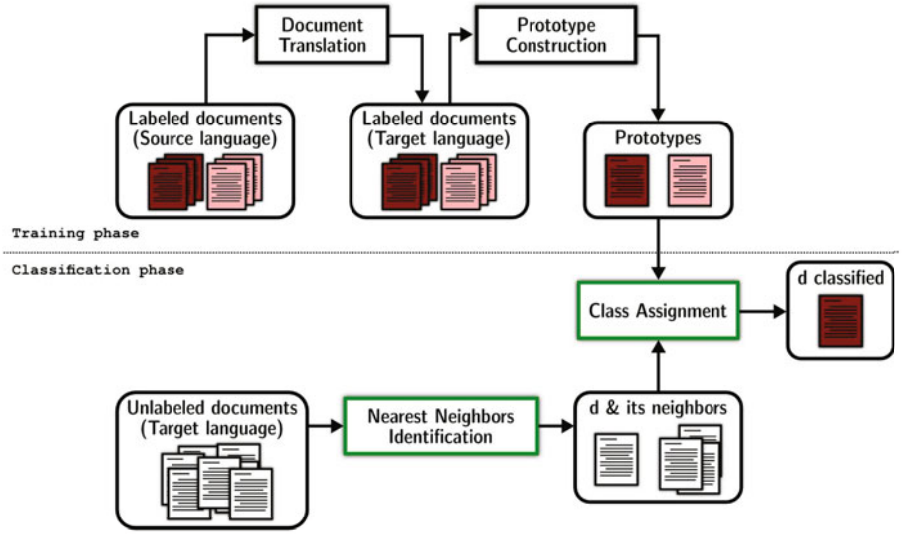


Fig. 1. General scheme of the proposed text classification method

Document Translation. Two basic architectures have been explored for CLTC, one based on the translation of the target dataset to the source language, and another one based on the translation of the training set to the target language. We decided to adopt the latter option because training sets are commonly smaller than test sets and, therefore, their translation tend to be less expensive. In particular, the translation was achieved using the Worldlingo online translation machine[1].

Prototype Construction. This process carries out the construction of the class prototypes based on information from the –translated– training set; thus, the resulting prototypes are represented in the target language. In particular, given a set $D = \{d_1, d_2, ...\}$ of vectors of labeled documents (from the training

[1] http://www.worldlingo.com/es/products_services/worldlingo_translator.html

set) organized in a predefined set of classes C and represented in their own term space, it computes the prototype vector for each class $c_i \in C$ using Formula 1.

$$P_i = \frac{1}{||\sum_{d \in c_i} d||} \sum_{d \in c_i} d \tag{1}$$

Nearest Neighbors Identification. This process focuses on the identification of the k nearest neighbors for each document d_i from the target dataset D_T (refer to Formula 2). In order to do that we compute the similarity between two documents (d_i and all other d in D_T) using the cosine formula (refer to Formula 4).

$$N_k^{d_i} = argmax_{S_j \in \mathbb{S}_k} \left[\sum_{d \in S_j} sim(d, d_i) \right] \tag{2}$$

where $\mathbb{S}_k$ and $sim()$ are defined as follows:

$$\mathbb{S}_k = \{S | S \subseteq D_T \wedge |S| = k\} \tag{3}$$

$$sim(d_i, d_j) = \frac{d_i \cdot d_j}{||d_i|| \times ||d_j||} \tag{4}$$

Class Assignment. In prototype-based classification, the class of a document d from the target dataset is traditionally determined by Formula 5. Our proposal extends this class-assignment strategy by considering not only information from the document itself but also information about the assigned category to other similar documents from the same target dataset. In particular, given a document from the target dataset ($d \in D_T$) in conjunction with its k nearest neighbors (N_k^d), we assign a class to d using Formula 6.

$$class(d) = argmax_i \left(sim(d, P_i) \right) \tag{5}$$

$$class(d) = argmax_i \left(\lambda \, sim(d, P_i) + (1 - \lambda) \frac{1}{k} \sum_{n_j \in N_k^d} [inf(d, n_j) \times sim(n_j, P_i)] \right) \tag{6}$$

where,

- $sim(v_i, v_j)$ is the cosine similarity function defined in Formula 4.
- N_k^d is the set of k neighbors considered to provide information about document d (refer Formula 2).
- λ is a constant used to determine the relative importance of both, the information from the document (d) and the information from its neighbors. The smaller the value of λ is, the greater the contribution of the neighbors, and vice versa.

– $inf()$ is an influence function used to weight the contribution of each neighbor n_j to the classification of d. The purpose of this function is to give more relevance to the closer neighbors. In particular, we define this influence in direct proportion to the similarity between each neighbor and d calculated using the cosine formula (refer to Formula 4).

3 Evaluation

3.1 Datasets

For the experiments we considered a subset of the Reuters RCV-1 Corpus [9]. This subset considers three languages (English, French and Spanish), and the news reports corresponding to four classes (Crime, Disasters, Politics, and Sports). For each language we used 320 documents; 80 per each class[2].

3.2 Evaluation Measure

The evaluation of the performance of the proposed method was carried out by means of the F-measure. This measure is a linear combination of the precision and recall values from all class $c_i \in C$. It is defined as follows:

$$F - Measure = \frac{1}{|C|} \sum_{i=1}^{|C|} \left[\frac{2 \times Recall(c_i) \times Precision(c_i)}{Recall(c_i) + Precision(c_i)} \right] \tag{7}$$

$$Recall(c_i) = \frac{number\ of\ correct\ predictions\ of\ c_i}{number\ of\ examples\ of\ c_i} \tag{8}$$

$$Precision(c_i) = \frac{number\ of\ correct\ predictions\ of\ c_i}{number\ of\ predictions\ as\ c_i} \tag{9}$$

3.3 Baseline Experiments

The goal of these experiments was to evaluate the performance of a traditional CLTC approach, where documents from a source language are used to classify documents from a different target language. For these experiments we applied the following standard procedure: first, we translated the training documents from the source language to the target language (using Worldlingo); then, we constructed a classifier (in the target language) using the translated training set; finally, we used the built classifier to determine the class of each document from the target-language dataset. For the construction of the classifier we considered three of the most used methods for text classification, namely, Naïve Bayes (NB), Support Vector Machines (SVM)[3], and a prototype-based method (PBC)

[2] This corpus can be downloaded from http://ccc.inaoep.mx/∼mmontesg/resources/CLTC/RCV-Subset.txt

[3] For NB and SVM we used the implementation and default configuration of WEKA [10].

Table 1. F-measure results for six crosslingual experiments using a traditional CLTC approach

Source language	Target language	Experiment	PBC	NB	SVM
English	French	$E_F - F$	0.616	0.753	0.764
English	Spanish	$E_S - S$	0.814	0.791	0.625
French	English	$F_E - E$	0.956	0.931	0.616
French	Spanish	$F_S - S$	0.879	0.882	0.658
Spanish	English	$S_E - E$	0.851	0.891	0.486
Spanish	French	$S_F - F$	0.790	0.802	0.723

using the class-assignment function described in Formula 5. Table 1 shows the F-measure results obtained by these methods in six crosslingual experiments, which correspond to all possible pair-combinations of the three selected languages. From these results, those by PBC are of special interest since our method is an extension of this approach.

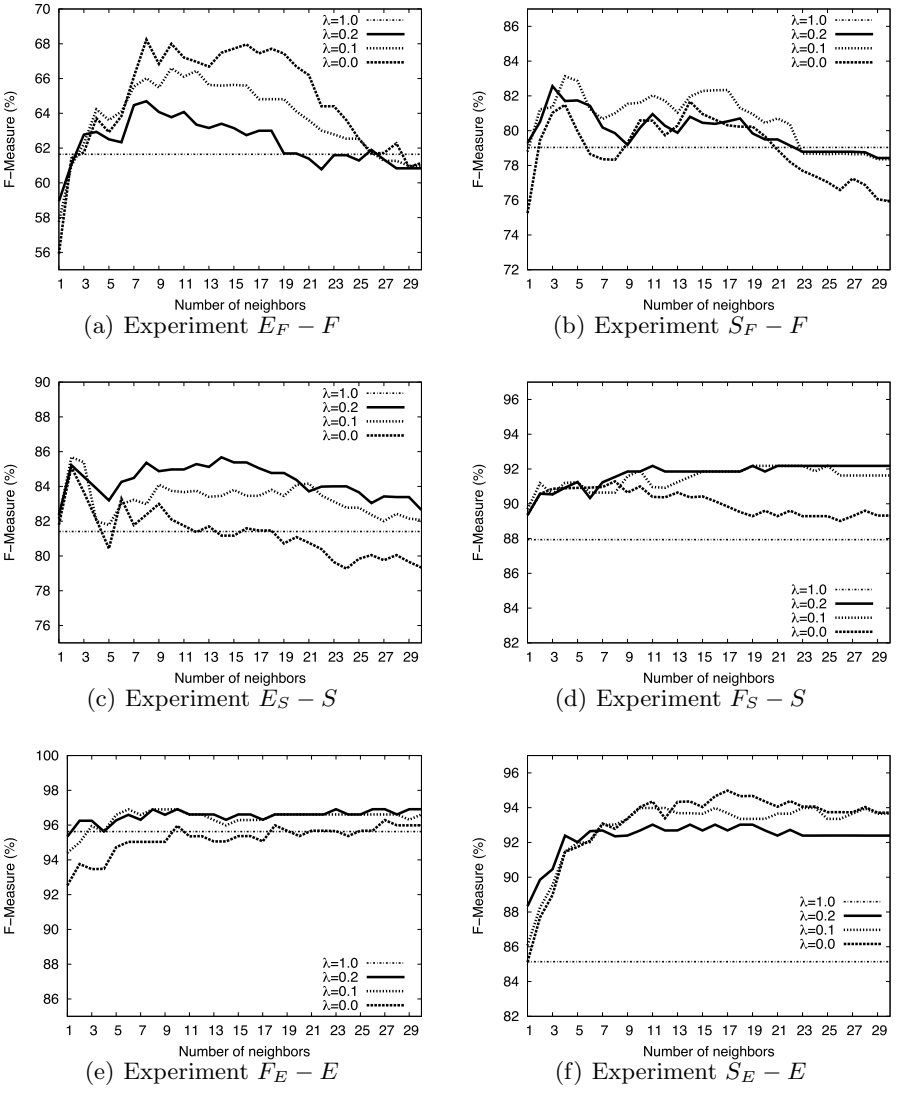
3.4 Results from the Proposed Method

As described in Section 2, the main idea of the proposed method is to classify the documents by considering not only their own content but also information from other similar documents from the same target dataset. Particularly, we adapted the traditional prototype-based approach (PBC) to capture this information (refer to Formula 6), being λ a constant that determines the relative importance of both components.

Considering the proposed method, we designed some experiments in such a way that we could evaluate the impact on the classification results caused by the selection of different values of λ, as well as the impact caused by the usage of different number of neighbor documents into the class assignment process. In particular, we used $\lambda = 0, 0.1, 0.2, ..., 1$, and $k = 1, ..., 30$.

Experiments showed that the best results were achieved when using small values of λ, indicating that information from the neighbor documents is of great relevance. On the other hand, they could not indicate a clear conclusion about the appropriate number of neighbors, since several different values allowed to obtain similar classification improvements. Figure 2 shows some results of the proposed method in the six crosslingual experiments. These results correspond to three different values of λ: 0, 0.1 and 0.2. This figure also shows the results from the traditional prototype-based approach, which correspond to our method results using $\lambda = 1$. The achieved results indicate that the proposed method clearly outperforms the traditional prototype-based approach.

In order to summarize the results from the experimental evaluation, Table 2 presents the best results achieved by the proposed method. Comparing these results against those from Table 1, it is possible to notice that our method outperformed all used classification algorithms in all except one of the crosslingual



(a) Experiment $E_F - F$ (b) Experiment $S_F - F$

(c) Experiment $E_S - S$ (d) Experiment $F_S - S$

(e) Experiment $F_E - E$ (f) Experiment $S_E - E$

Fig. 2. F-measure results of the proposed method in the six crosslingual experiments, using different values of λ and numbers of neighbors (k). The straight line corresponds to the PBC baseline result ($\lambda = 1$).

experiments, demonstrating the usefulness of considering information from the target dataset in crosslingual text classification.

At this point it is important to clarify that several different configurations of our method (as shown in Figure 2) allowed obtaining competitive classification results. One example is the configuration defined by $\lambda = 0.1$ and $k = 11$, which

Table 2. Best F-measure results of the proposed method

Experiment	Baselines		Best results		Configuration
	PBC⋆	Best†	[k, λ]		[$k = 11$, $\lambda = 0.1$]
$E_F - F$	0.616	-	0.682 ⋆	[8, 0.0]	0.661
$E_S - S$	0.814	0.814	0.857 ⋆†	[2, 0.1]	0.837
$F_E - E$	0.956	-	0.969	[10, 0.2]	0.966
$F_S - S$	0.879	0.882	0.922 ⋆†	[11, 0.2]	0.910
$S_E - E$	0.851	0.891	0.950 ⋆†	[17, 0.0]	0.940
$S_F - F$	0.790	-	0.831 ⋆	[4, 0.1]	0.820

also outperformed most baseline results as shown in the last column of Table 2. We evaluated the statistical significance of the best achieved results using the z-test with a confidence of 95%; a ⋆ indicates that the improvement over the PCB is statistically significant, whereas, a † indicates the same regarding the best baseline result.

4 Conclusions and Future Work

In addition to the evident translation problem, crosslingual text classification (CLTC) also faces some difficulties caused by the cultural discrepancies manifested in both languages by means of different topic distributions. In this paper we proposed a simple and inexpensive approach for facing this problem. This approach is based on the idea that similar documents must belong to the same category and, therefore, it classifies the documents by considering their own information (as usual) as well as the information about the assigned category to other similar documents.

In particular, we implemented the proposed approach using the prototype-based classification algorithm. In our implementation the decision about the category of each document (from the target language) is determined by the class whose prototype (calculated from the training set) is more similar to it and to its nearest neighbors (from the same target-language dataset). This way, the proposed method determines the category of documents taking advantage of information from the two languages.

As future work we plan to carry out an extensive analysis of several crosslingual experiments (using different languages and a larger number of documents) to establish a simple criterion for determining the appropriate values for parameters λ and k. Once defined this criterion, we also plan to use the proposed approach in conjunction with a semi-supervised method as the one described by Rigutini et al. [4]. Our goal is to enhance the selection of the documents that will be iteratively included in the training set, and, consequently, to obtain a classification model that is as close as possible to the target-language distribution.

Acknowledgments. This work was done under partial support of CONACyT-Mexico (project grants 83459, 82050, 106013 and 106625, and scholarship 239516).

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34, 1–47 (2002)
2. Bel, N., Koster, C.H.A., Villegas, M.: Cross-lingual text categorization. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 126–139. Springer, Heidelberg (2003)
3. de Melo, G., Siersdorfer, S.: Multilingual text classification using ontologies. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 541–548. Springer, Heidelberg (2007)
4. Rigutini, L., Maggini, M., Liu, B.: An EM based training algorithm for cross-language text categorization. In: WI 2005: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, pp. 529–535. IEEE Computer Society, Los Alamitos (2005)
5. Ling, X., Xue, G.R., Dai, W., Jiang, Y., Yang, Q., Yu, Y.: Can Chinese web pages be classified with English data source? In: WWW 2008: Proceeding of the 17th International Conference on World Wide Web, pp. 969–978. ACM, New York (2008)
6. Wan, X.: Co-training for cross-lingual sentiment classification. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore, Association for Computational Linguistics, pp. 235–243 (2009)
7. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
8. Cardoso-Cachopo, A., Oliveira, A.L.: Semi-supervised single-label text categorization using centroid-based classifiers. In: SAC 2007: Proceedings of the 2007 ACM Symposium on Applied Computing, pp. 844–851. ACM, New York (2007)
9. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. J. Mach. Learn. Res. 5, 361–397 (2004)
10. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Event Detection Using Lexical Chain

Sangeetha S.[1], R.S. Thakur[2], and Michael Arock[3]

[1] Department of Computer Applications, National Institute of Technology, Tiruchirapalli–620015, Tamilnadu, India
sangeetha@nitt.edu
[2] Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India
ramthakur2000@yahoo.com
[3] Department of Computer Applications, National Institute of Technology, Tiruchirapalli–620015, Tamilnadu, India
michael@nitt.edu

Abstract. This paper describes a new architecture for event detection from text documents. The proposed system correctly identifies the sentences that describe an event of interest to extract its participants. It follows an unsupervised method for identifying the lexical chains from the raw sentences taken as a training data. The lexical chain constructed using Wordnet lexicon is then used for identifying event mention. The significance of the proposed system is it is the first system that applies lexical chain for event identification. The entire architecture is divided into three tasks namely, natural language pre-processing, lexical chain construction and event detection.

Keywords: Event Extraction, Lexical chain.

1 Introduction

Information extraction (IE) [1] is the process of extracting the structured information from the unstructured text. IE systems were evaluated by Message Understanding Conferences (MUC) till 1998. Automatic content extraction (ACE) programme is the successor of MUC with the objective of developing extraction technology to support automatic processing of source language data.

Event identification and characterization of ACE [3] programme identifies events by making use of event triggers. Event trigger is the word that clearly expresses the occurrence of an event. Event mention is the sentence in which the event is described. An event comprises event participants,which are the entities that participate in the event with different roles.

As the supervised machine learning requires a large set of event annotated data, our approach uses unsupervised method for extracting the events. The proposed method constructs the lexical chain from the un-annotated or raw sentences of training documents. Section 4 explains how the training documents are constructed. Lexical chain holds the set of semantically related words of a given sentence or a document from which it was obtained. Word net lexicon [15] is used for constructing lexical chain in the proposed work. The significance of

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 314–319, 2010.

the proposed system is, it is the first system that uses lexical chain for event extraction to the best of our knowledge. The entire architecture is divided into three tasks namely, natural language pre-processing, lexical chain construction and event detection.

The remaining sections of the paper are organized as follows. The next section describes the related work in the field of information extraction generally and event extraction specifically. Section 3 provides a description of proposed approach, Section 4 explains how the training documents are constructed and partial experimental results and Section 5 concludes the paper.

2 Related Work

McCracken et al. [2] had combined statistical and knowledge based technique for extracting events. Its main focus was on summary report genre. It had extracted the factual accounting of incidents from a person's life. This resource had covered every instance of every verb in the corpus. Xu et al. [5] had developed a method for identifying event extent, event trigger and event argument automatically using bootstrapping method. The work had extracted the events from the Nobel Prize winning domain by obtaining extraction rules from the text fragments using binary relations as seeds.

Abuleil [4] proposed a method which extracted events by breaking each event into elements, analysed and understood the syntax of each element, identified the role played by each element in the event and how they formed relationship between related events. David Ahn [7] broke down the task of extracting the events into subtasks such as anchor identification, argument identification, attribute assignment and event co-reference. Each task was performed with the help of machine learned classifier. Some of the learners used are memory based learners and maximum entropy learners.

Aone et al. [6] has identified events by tagging the text with name and noun phrase using pattern matching techniques and has resolved co-references using rule based approach. It did not consider the semantic relatedness.

All the above said existing systems extract the events without considering the semantic features of the text. However, consideration of meaning of the text improves the efficiency of event extraction, and the information extraction on the whole. The actual meaning of a sentence is identified from the meaning of individual words. Our approach uses lexical chain for identifying the events because lexical chain holds the semantically related words.

The concept, lexical chain [8] is based on the cohesion [9] which is a method for sticking different part of the text which is semantically related. Lexical chaining has been used in various tasks such as text summarization [12], word sense disambiguation and web information retrieval [11]. Naughton et al. [10] stated that event identification using manual extraction of trigger terms from Lexicon such as Wordnet performs well. Our approach improves this trigger by adding some more related words of the same concept by constructing lexical chain. Naughton et al. [10] manually extracted the terms related to an event type. In our approach

we learn all the words used in the sentences which represent an event and also the related words from Wordnet semantic relations. This includes more related words and will improve the result.

From the literature we have identified that (a) lexical chain holds set of semantically related words, (b) to identify an event we need the words related to a particular event and also (c) to the best of our knowledge, there is no event extraction system that uses lexical chain so far. Hence we propose an architecture which identifies an event by constructing LC based on training sentences.

3 Proposed Work

The proposed architecture is broadly classified into three phases (Fig. 1) namely preprocessing, lexical chain construction and event detection.

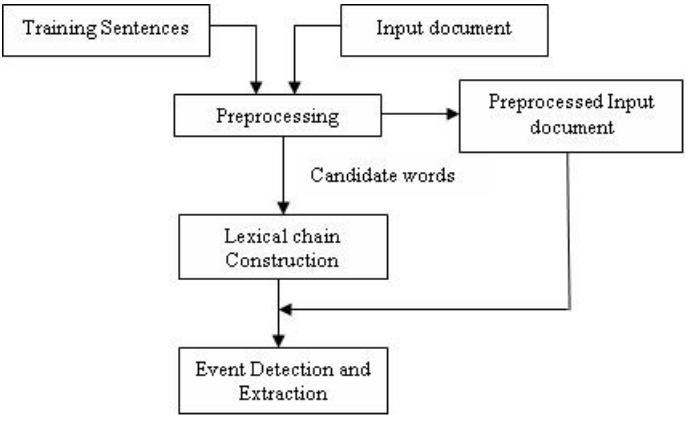


Fig. 1. Architecture Overview

Pre-processing

Preprocessing includes part of speech tagging, stop-word removal and named entity identification. Collect the sentences describing a particular event of interest to form a single document and pass them as input to the preprocessing stage.

We have adopted standard tree bank POS tagger for part of speech tagging and ACE named entity chunking available in [14]. At the end of preprocessing stage we obtain the tokenized tagged words from the training sentences along with the types of named entities.

Lexical Chain Construction

The preprocessing phase produces output as a list of words collected from all the training sentences. These words are the candidate words for constructing lexical chain. Several algorithms have been proposed for lexical chain. We have used the algorithm developed by [8] with some enhancement, using the Word net

lexicon. Moriss and Hirst constructed lexical chain which includes only the words available in the text which are semantically related. In our approach, along with the words in the text, we have included the words in the Word net lexicon that are used to establish the semantic relation between any two words in the text. Moreover, the proposed approach includes noun, verb and adjective words, as all these POS can act as an event trigger according to ACE [3]. Since we are using the words in the lexical chain as a trigger word for identifying the event, our procedure learns not only the semantically related words from the training data but also the related words from the Word net lexicon. The procedure for constructing the lexical chain is given below:

All the senses of each candidate words are extracted. Each sense of the word is represented by the set of words in its synonym at 0^{th} level, and hypernym/hyponym,meronym/holonym at level 1. For each pair of candidate words, each sense of the word W_i is compared with all other senses of word W_j. If the match occurs, the words W_i, $W_j$and the list of matched words in the sense representation are included in the chain. Along with the matched words, path length with respect to the current level of matching and frequency of the word are also stored. The frequent words with minimum path length are included in the lexical chain. The list of all words forms overall LC. From the set of LC's formed we manually select the LC which is related to the event. As the sentences in the training documents are related to a particular event the required LC is always longer than other LC's constructed.

Event Detection and Extraction
In a given document, the sentences are searched for words in lexical chain. Sentence score is calculated based on how many words of LC it contains. The sentences with highest score are the sentences representing the type of event. This method can be adopted to any type of event by selecting the training sentences denoting the particular event type. We mark the event mention as "**relevant**" if its score is above threshold. More number of words in the lexical chain reflects the large set of words related to the event, which in turn identify large set of event mentions. As a result the final precision and recall will improve. Keeping this in mind we are proceeding our research and obtained initial set of results in Fig. 2, which shows how redundancy in training document affects the number of words in the lexical chain.

To extract the event participants, the event mention should be tokenized, POS tagged, stemmed and named entities should be recognized. From the lexical chains, patterns using preposition and type of named entities are identified to extract the required fields. Then subject and object of the sentence are identified which denote the agent and theme of the event which can be mapped to event participants.

4 Experiment and Discussions

We have collected 50 documents representing the education and administrative positions held by various international leaders from Wikipedia [13]. The documents

are chosen in such a way that it uses different words to represent education and administrative positions details in each document. Among the 50 documents we have used 25 documents for training and remaining 25 documents for testing. The sentences in the training documents are manually separated into sentences describing education details and sentences describing details of administrative positions held and they are grouped under two different documents. These documents are used as input to the preprocessing stage. In our experiment, first we have considered the document with the sentences related to education.

This system is under development. Till now, we have completed the construction of the lexical chain. In this paper we illustrate the preliminary result of our proposed architecture. If the number of words in lexical chain is greater, we can identify the wide range event mentions which will improve the final precision and recall value. Thus importance should be given to the number of words in the lexical chain. Redundancy of context words in the training document is the main factor which is affecting the number of words in LC. Our experimental result confirms it. From the empirical analysis, we have identified that if candidate words are less redundant in the same context, the number of words in the lexical chain increases. Whereas, if redundancy is introduced in the context words of the training document, number of words in the lexical chain decreases. Fig. 2 shows it as the preliminary result.

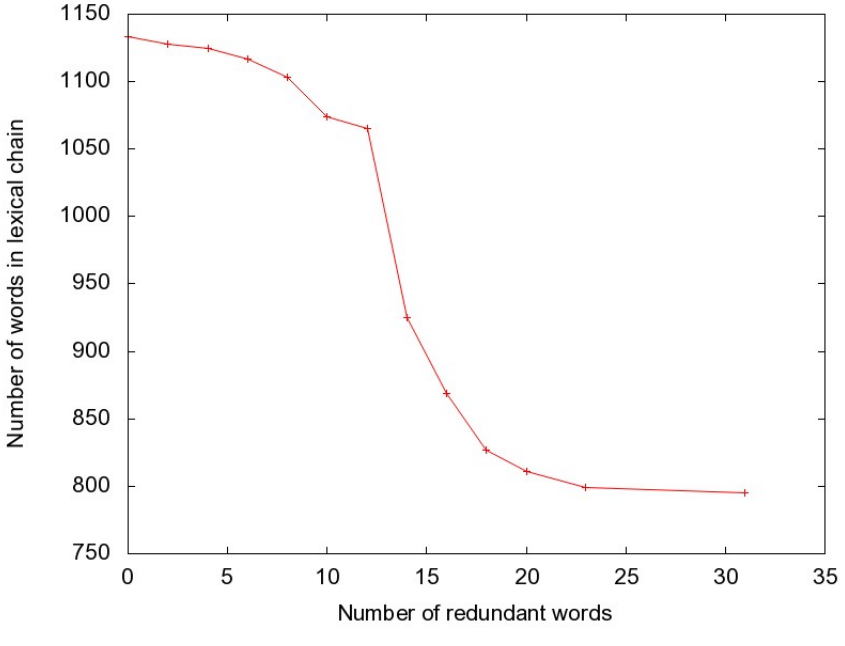


Fig. 2. Lexical chaining vs redundancy in context words

5 Conclusion

The proposed work provides an architecture that extracts the events by using lexical chains. Most of the existing systems extract the events without considering the semantics features of the text. However, consideration of semantics of the text improves the efficiency of event extraction, and the information extraction on the whole. Our approach identifies the semantic relationship between words and constructs lexical chains based on the relationship. Lexical chain is used to identify the event mention within the documents as "**relevant**" or "**irrelevant**" based on the sentence score. From our empirical results we conclude that less redundant training sentences will acquire more number of unique words in the lexical chain and large set of words in lexical chain correctly identify large set of event mentions.

References

1. Cunningham, H.: Information Extraction. Automatic. Encyclopaedia of Language and Linguistics, 665–677 (2005)
2. McCracken, N., Ozgencil, N.E., Symonenko, S.: Combining Techniques for Event Extraction in Summary Reports. In: AAAI 2006 Workshop Event Extraction and Synthesis, pp. 7–11 (2006)
3. ACE (Automatic Content Extraction) English Annotation Guidelines for Events Version 5.4.3 2005.07.01 Linguistic Data Consortium, http://www.ldc.upenn.edu
4. Abuleil, S.: Using NLP techniques for Tagging Events in Arabic Text. In: 19th IEEE International Conference on Tools with AI, pp. 440–443. IEEE Press, Los Alamitos (2007)
5. Xu, F., Uszkoreit, H., Li, H.: Automatic Event and Relation Detection with Seeds of Varying Complexity. In: AAAI 2006 Workshop Event Extraction and Synthesis, Boston, pp. 491–498 (2006)
6. Aone, C., Ramos-Santacruz, M.: REES: a large-scale relation and event extraction system. In: Sixth Conference on Applied Natural Language Processing, pp. 76–83. Morgan Kaufmann Publishers Inc., Washington (2000)
7. David, A.: Stages of event extraction. In: COLING/ACL 2006 Workshop on Annotating and Reasoning about Time and Events, ACL, pp. 1–8 (2006)
8. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics 17(1), 21–48 (1991)
9. Haliday, M., Hansan, R.: Cohesion in English. Longman, London (1976)
10. Naughton, M., Stokes, M., Carthy, J.: Investigating Statistical Techniques for sentence level event classification. In: Coling, pp. 617–624 (2008)
11. Hatch, P., Stokes, N., Carthy, J.: Lexical chains for web based retrieval of breaking news. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) AH 2000. LNCS, vol. 1892, pp. 327–330. Springer, Heidelberg (2000)
12. Brunn, M., Chali, Y., Pinchak, C.J.: Text Summarization Using Lexical Chains. In: Proceedings of the Document Understanding Conference, NIST, pp. 135–140 (2001)
13. http://www.wikipedia.org
14. http://www.nltk.org
15. Miller, G.: Wordnet: A lexical database for English. Communications of the ACM 38(11), 39–41 (1995)

Using Comparable Corpora to Improve the Effectiveness of Cross-Language Information Retrieval

Fatiha Sadat

University of Quebec in Montreal, Computer Science department,
201 President Kennedy avenue, Montreal, QC, Canada
sadat.fatiha@uqam.ca

Abstract. Large-scale comparable corpora became more abundant and accessible than parallel corpora, with the explosive growth of the World Wide Web. From the Cross-Language Information Retrieval point of view, limitation of translation resources as well as ambiguity arising due to failure to translate query terms is largely responsible for large drops in the effectiveness below monolingual performance. Therefore, strategies on bilingual terminology extraction from comparable texts must be given more attention in order to enrich existing bilingual lexicons and thesauri and to enhance Cross-Language Information Retrieval. In the present paper, we focus on the enhancement of Cross-Language Information Retrieval using a two-stage corpus-based translation model that includes bi-directional extraction of bilingual terminology from comparable corpora and selection of best translation alternatives on the basis of their morphological knowledge. The impact of comparable corpora on the performance of the Cross-Language Information Retrieval process is evaluated in this study and the results indicate that the effect is clearly positive, especially when using the linear combination with bilingual dictionaries and Japanese-English pair of languages.

Keywords: Cross-language information retrieval, comparable corpora, similarity, translation, disambiguation.

1 Introduction

Cross-Language Information Retrieval (CLIR) deals with the problem of presenting an information retrieval task in one language and retrieving documents in one or several other languages. The main methods for CLIR are presented in overviews by Oard & Diekema (1998), and Pirkola et al. (2001). Methods based on the translation of queries are categorized into either (i) dictionary-based translation, (ii) machine translation, (iii) methods using parallel corpora and other approaches, which are based on other existing linguistic resources, such as the thesaurus. However, according to previous research (Sadat et al., 2002) the main problems, which are largely responsible for large drops in the effectiveness of CLIR below monolingual performance, are listed as follows: First is the problem of inflection. A commonly used method to deal with inflected words is to remove affixes from word forms. The method is called stemming. Morphological analysis also allows the normalization of words forms into

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 320–331, 2010.

their base forms. The second problem associated to CLIR is related to the translation ambiguity arising from polysemous words. Third is the problem of compounds, phrases, multi-words (e.g. specialized vocabulary) and their handling and failure of translation. Compound words form an important part of natural language, since compounding is a major way of forming new words. From the information retrieval (IR) point of view, compounds may be content bearing words in natural language sentences and therefore important for the retrieval result (Hedlund, 2002). The problem with compound handling for CLIR is acknowledged for many languages. The forth problem encountered in CLIR is related to proper names, named entities and other untranslatable words using existing translation tools (bilingual dictionary and/or machine translation). Lexical coverage of existing bilingual dictionary, limitation of general-purpose dictionaries especially for specialized vocabulary and inexistence of translation tools for pairs of languages are among the merging problems that could be solved using large-scale corpora.

In recent years two types of multilingual corpora have been an object of studies and research related to natural language processing and information retrieval: parallel corpora and comparable corpora. The parallel corpora are made up of original texts and their translations. This allows texts to be aligned and used in applications such as computer-aided translator training and machine translation systems. This method could be expensive for any pair of languages or even not applicable for some languages, which are characterized by few amounts of Web pages on the Web. On the other hand, non-aligned comparable corpora, more abundant and accessible resources than parallel corpora, have been given a special interest in bilingual terminology acquisition and lexical resources enrichment (Dagan & Itai, 1994; Dejean et al., 2002; Diab & Finch, 2000; Fung, 2000; Gaussier et al., 2004; Kaji, 2003; Koehn & Graehl, 2002; Nakagawa, 2000; Peters & Picchi, 1995; Rapp, 1999; Sadat et al., 2003a; Sadat et al., 2003b; Sadat et al., 2003c; Sadat, 2004; Shahzad et al., 1999; Tanaka & Iwasaki, 1996; Utsuro et al., 2002 ; Utsuro et al., 2003). Comparable corpora are defined as collections of texts from pairs or multiples of languages, which can be contrasted because of their common features in the topic, the domain, the authors, the time period, etc. Comparable corpora could be collected from downloading electronic copies of newspapers and articles, on the WWW for any specified domain.

This paper intends to bring solutions to the problem of lexical coverage of existing bilingual dictionaries but also to the improvement of the performance of CLIR. The main contributions concern the enhancement of CLIR by an automatic acquisition of bilingual terminology from comparable corpora that will help cope with the limitation of CLIR, especially in the query disambiguation process as well as during the query expansion with related terms. Furthermore, this study could be valuable for the extraction of unknown words and their translation and thus the enrichment and enhancement of bilingual dictionaries. Therefore, we present in this paper an approach of learning bilingual terminology from textual resources other than bilingual dictionaries, such as comparable corpora and evaluations on CLIR. First, we propose a two-stage corpus-based translation model for the acquisition of bilingual terminology from comparable corpora. The first stage concerns the extraction of bilingual translations from the source language to the target language, also from the target language to the source language. The two results are combined for the purpose of disambiguation. In the second stage, the extracted translation alternatives are filtered on the basis of their

morphological knowledge. A linguistics-based pruning technique is applied in order to compare source words and their target language translation equivalents on the basis of their part of speech tags. Furthermore, we present a combined translation model involving the comparable corpora and readily available bilingual dictionaries. In our evaluations, we used a large-scale test collection on Japanese-English and different weighting schemes of SMART retrieval system and confirmed the effectiveness of the proposed translation model in CLIR.

The remainder of the present paper is organized as follows: Section 2 presents an overview of the proposed model. Section 3 presents the two-stage corpus-based translation model. Section 4 introduces a combination of different translation models. Experiments and evaluations in CLIR are related in Section 5. Section 6 concludes the present paper.

2 An Overview of the Proposed Model

Throughout this paper we will seek to exploit and explore benefits from collections of news articles for the acquisition of bilingual terminology, in order to enrich existing multilingual lexical resources and help cross the language barrier for information retrieval. We rely on such comparable corpora for the extraction of bilingual terminology, in the form of translations and/or expansion terms, i.e. words that will help the query expansion in CLIR. First, a linguistic preprocessing is performed on the comparable corpora in order to replace each term with its inflectional root, to remove most plural word forms, to replace each verb with its infinitive form, to remove stop words and stop phrases and finally to extract content words, such as nouns, verbs, adjectives, adverbs and foreign words, which will constitute the main target of our study. Second, the task of bilingual terminology extraction is accomplished by a two-stage corpus-based translation model, which is described in detail in Section 3. Third, a linear combination involving the comparable corpora and bilingual dictionaries is completed in order to select best translation candidates of the source terms of a given query. Finally, documents are retrieved in the target language.

3 Two-Stage Corpus-Based Translation Model

A two-stage corpus-based translation model (Sadat et al., 2003a; Sadat et al., 2003b; Sadat et al., 2003c), which is based on the symmetrical criterion in addition to the assumption of similar collocation, aims to find translations of the source word in the target language corpus but also translations of the target words in the source language corpus. Linguistic resources were used in the two-stage corpus-based translation model, as follows: (i) a collection of news articles from *Mainichi Newspapers* (1998-1999) for Japanese and *Mainichi Daily News* (1998-1999) for English were considered as comparable corpora, because of their common feature on the time period. Documents of *NTCIR-2* test collection were also considered as comparable corpora in order to cope with special features of the test collection during evaluations; (ii) morphological analyzers, *ChaSen* version 2.2.9 (Matsumoto et al., 1997) for texts in Japanese and *OAK* (Sekine, 2001) for English texts were used in linguistic processing; (iii)

EDR (1996) and EDICT[1] bilingual Japanese-English and English-Japanese dictionaries were considered in the translation of context vectors of source and target languages. Japanese words written in Katakana representing foreign words and proper names, that were not found in the bilingual dictionaries were manually translated. A transliteration process could be used in order to convert those words to their English equivalence.

3.1 First Stage in the Proposed Translation Model

The two-stage corpus-based translation model for the acquisition of bilingual terminology is described as follows:

1. A simple bilingual terminology acquisition from source language to target language to yield a first simple translation model represented by similarity vectors $SIM_{S \to T}$.
2. A simple bilingual terminology acquisition from target language to source language to yield a second simple translation model represented by similarity vectors $SIM_{T \to S}$.
3. Merge the first and second models to yield a two-stage translation model, based on bi-directional comparable corpora and represented by similarity vectors $SIM_{S \leftrightarrow T}$.

The simple approach for bilingual terminology acquisition from comparable corpora is based on the assumption of similar collocation, i.e., If two words are mutual translations, then their most frequent collocates are likely to be mutual translations as well. We follow strategies of previous researches (Dejean et al., 2002; Fung, 2000; Rapp, 1999; Sadat et al., 2003a; Sadat et al., 2003b, Sadat et al., 2003c). The approach is described as follows: First, word frequencies, context word frequencies in surrounding positions (here three-words window) are estimated following statistics-based metrics. Context vectors for each term in the source language are constructed. As well, context vectors for terms in the target language are constructed. We use the *log-likelihood ratio* (Dunning, 1993) for the estimation of context frequencies for each pair of words in either the source language or target language, as expressed in equation (1).

$$LLR(w_i, w_j) = K_{11} log \frac{K_{11}N}{C_1R_1} + K_{12} log \frac{K_{12}N}{C_1R_2} + K_{21} log \frac{K_{21}N}{C_2R_1} + K_{22} log \frac{K_{22}N}{C_2R_2} \quad (1)$$

where, $C_1 = K_{11} + K_{12}$, $C_2 = K_{21} + K_{22}$, $R_1 = K_{11} + K_{21}$, $R_2 = K_{12} + K_{22}$, $N = K_{11} + K_{12} + K_{21} + K_{22}$,, K_{11} = frequency of common occurrences of word w_i and word w_j in a specified window size of the monolingual corpus, K_{12} = corpus frequency of word w_i, in the corpus - K_{11}, K_{21} = corpus frequency of word w_j in the corpus - $K_{11,}$, $K_{22} = N - K_{11} - K_{12} - K_{21}$.

Second, context vectors of words in the source language are translated into the target language using a bilingual seed lexicon. We consider all translation candidates, keeping the same context frequency value as the source word. This step requires a

[1] http://www.csse.monash.edu.au/~jwb/wwwjdic.html

seed lexicon that will be enriched using the proposed bootstrapping approach of this paper.

The third step is the construction of similarity vectors for each pair of source word and target word. Context vectors (original and translated) of words in both languages are compared using the *cosine metrics* (Salton & McGill, 1983) as expressed in equation (2). Finally, similarity vectors are normalized to yield a simple corpus-based translation model.

$$Similarity\ (w_i, w_j) = \frac{\sum_k v_{ik} v_{jk}}{\sqrt{\sum_k v_{ik}^2 \sum_k v_{jk}^2}} \tag{2}$$

where,

v_{ik} represents co-occurrence frequencies of the source word w_i with word w_k. The word w_k is found in the translated context vectors of the source word w_i.

v_{jk} represents co-occurrence frequencies of the target word w_j with the word w_k. The word w_k is found in the context vectors of the target word w_j.

The merging strategy in the first stage of the two-stage corpus-based translation model is represented by the following equation (3):

$$\begin{aligned} \mathrm{SIM}_{S\leftrightarrow T} = \{&(s,t,sim_{S\leftrightarrow T}(t\,|\,s))/(s,t,sim_{S\to T}(t\,|\,s)) \in SIM_{S\to T} \wedge (t,s,sim_{T\to S}(s\,|\,t)) \in SIM_{T\to S} \\ &\wedge sim_{S\leftrightarrow T}(t\,|\,s) = sim_{S\to T}(t\,|\,s) \times sim_{T\to S}(s\,|\,t)\} \end{aligned} \tag{3}$$

Similarity vectors $SIM_{S\to T}$ and $SIM_{T\to S}$ for the first and second models are constructed and merged to yield a bi-directional acquisition of bilingual terminology from source language to target language. The merging process will keep common pairs of source term and target translation *(s,t)* which appear in $SIM_{S\to T}$ as pairs *(s,t)* but also in $SIM_{T\to S}$ as pairs *(t,s)*, to result in combined similarity vectors $SIM_{S\leftrightarrow T}$ for each pair *(s,t)*.The product of similarity values $sim_{S\to T}(t|s)$ and $sim_{T\to S}(s|t)$ of vectors $SIM_{S\to T}$ and $SIM_{T\to S}$ consecutively, will result in similarity values $sim_{S\leftrightarrow T}(t|s)$ of vectors $SIM_{S\leftrightarrow T}$, which will represent the first stage of the two-stage corpus-based translation model. In further sections, we name the simple approach for bilingual terminology acquisition from comparable corpora as s*imple corpus-based translation* and the translation model representing the first stage of the two-stage corpus-based translation as *bi-directional corpus-based translation.*

3.2 Second Stage in the Proposed Translation Model

Combining linguistic and statistical methods is becoming increasingly common in computational linguistics, especially as more corpora become available (Klavens & Tzoukermann, 1996; Sadat et al., 2003c). We propose to integrate linguistic concepts into the corpus-based translation model. Morphological knowledge such as Part-of-Speech (POS) tags, context of terms, etc., could be valuable to filter and prune the extracted translation candidates. The objective of the linguistics-based pruning technique is the detection of terms and their translations that are morphologically close enough, i.e., close or similar POS tags. This proposed approach will select a fixed

number of equivalents from the set of extracted target translation alternatives that match the Part-of-Speech of the source term.

POS tags are assigned to each source term (Japanese) via morphological analysis. As well, a target language morphological analysis will assign POS tags to the translation candidates. We restricted the pruning technique to nouns, verbs, adjectives and adverbs, although other POS tags could be treated in similar way. For Japanese-English[2] pair of languages, Japanese nouns (MEISHI) are compared to English nouns (NN) and Japanese verbs (DOUSHI) to English verbs (VB). Japanese adverbs (FUKUSHI) are compared to English adverbs (RB) and adjectives (JJ); while, Japanese adjectives (KEIYOUSHI) are compared to English adverbs (RB) and adjectives (JJ). This is because most adverbs in Japanese are formed from adjectives. Thus, we select pairs of source term and target translation *(s,t)* such as:

POS(s) = "NN" and POS(t) = "MEISHI"
POS(s) = "VB" and POS(t) = "DOUSHI"
POS(s) = "RB" and [POS(t) = "FUKUSHI" or POS(t) = "KEIYOUSHI"]
POS(s) = "JJ" and [POS(t) = "KEIYOUSHI" or POS(t) = "FUKUSHI"]

Note that Japanese vocabulary is frequently imported from other languages, primarily (but not exclusively) from English. The special phonetic alphabet (here Japanese katakana) is used to write down foreign words, technical terms, proper nouns and loanwords, e.g. names of persons and entities. Japanese foreign words were not pruned with the proposed linguistics-based technique but could be treated via *transliteration*, i.e., conversion of Japanese katakana to their English equivalence or to the alphabetical description of their pronunciation (Knight & Graehl, 1998).

Finally, the generated translation alternatives are sorted in decreasing order by similarity values. Rank counts are assigned in increasing order, starting at 1 for the first sorted list item. A fixed number of top-ranked translation alternatives are selected and misleading candidates are discarded.

4 Combining Different Translation Models

Combining different translation models has showed success in previous research (Dejean et al., 2002).

We propose a combined translation model involving comparable corpora and readily available bilingual dictionaries. The proposed dictionary-based translation model is derived directly from readily available bilingual dictionaries, by considering all translation candidates of each source entry as equiprobable, to yield a probabilistic translation model $P_2(t|s)$. The linear combination will involve the two probabilistic translation models $P_1(t|s)$ and $P_2(t|s)$ derived from the comparable corpora (either the simple or the two-stage model) and readily available bilingual dictionaries, respectively as follows:

[2] English POS tags NN refers to noun, VB to verb, RB to adverb, JJ to adjective; while Japanese POS tags MEISHI refers to noun, DOUSHI to verb, FUKUSHI to adverb and KEIYOUSHI to adjective, with respect to their extensions.

$$P(t \mid s) = \sum_{\forall i} \alpha_i p_i(t \mid s)$$

Parameters α_1 and α_2 represent mixture weights of each translation source with $\sum_{\forall i} \alpha_i = 1$ Although, the mixture weights could be adjusted using the EM algorithm (Dejean et al., 2002), individual translation sources were assigned equiprobable weights, in these preliminary evaluations. Japanese vocabulary is frequently imported from other languages, primarily (but not exclusively) from English. *Katakana*, the special phonetic alphabet is used to write down foreign words and loanwords, example names of persons and other terms. A probabilistic translation model representing the transliteration could be integrated in the combined model as well.

5 Evaluation and Experiments

Experiments have been carried out in order to measure the improvement of our proposal on bilingual terminology acquisition from comparable corpora on Japanese-English tasks in CLIR, i.e. Japanese queries to retrieve English documents.

5.1 Evaluations on the Corpus-Based Translation Model

We considered the set of news articles as well as the abstracts of NTCIR-2 test collection as comparable corpora for Japanese-English language pairs. NTCIR2 contains abstracts from academic conference papers, with much technical terms that may not be found in the standard dictionaries or in general-domain news articles. The abstracts of NTCIR-2 test collection are partially aligned (more than half are Japanese-English paired documents) but the alignment was not considered in the present research in order to treat the set of documents as comparable. Content words (nouns, verbs, adjectives, adverbs and Foreign words) were extracted from English and Japanese corpora. Context vectors were constructed for 13,552,481 Japanese terms and 1,517,281 English terms. Similarity vectors were constructed for 96,895,255 (Japanese, English) pairs of terms and 92,765,129 (English, Japanese) pairs of terms. Bi-directional similarity vectors (after merging and disambiguation) resulted in 58,254,841 (Japanese, English) pairs of terms.

5.2 Evaluations on the Retrieval System

Conducted experiments and evaluations were completed using a large-scale test collection, *NTCIR-2* (Kando, 2001). *SMART* information retrieval system (Salton, 1971), which is based on vector model, was used to retrieve English documents. We used the monolingual English runs, i.e., English queries to retrieve English documents and the bilingual Japanese-English runs, i.e., Japanese queries to retrieve English documents. Topics of NTCIR-2 collection, numbered 0101 to 0149, were considered and key terms contained in the fields, title *<TITLE>*, description *<DESCRIPTION>* and concept *<CONCEPT>* were used to generate 49 queries in Japanese and in English. There is a variety of techniques implemented in SMART to calculate weights for individual terms in both documents and queries. These weighting techniques are formulated by combining three parameters: *Term Frequency* component, *Inverted*

Document Frequency component and *Vector Normalization* component. The standard SMART notation to describe the combined schemes is "XXX.YYY". The three characters to the left (XXX) and right (YYY) of the period refer to the document and query vector components, respectively. For example, ATC.ATN applies augmented normalized term frequency, *tf×idf* document frequency *(term frequency times inverse document frequency components)* to weigh terms in the collection of documents. Similarly ATN refers to the weighting scheme applied to the query.

First experiments were conducted on several combinations of weighting parameters and schemes of SMART retrieval system for documents terms and query terms, such as ATN, ATC, LTN, LTC, NNN, NTC, etc. In our experiments, best performances in terms of average precision were realized by the ATN.NTC combined weighting scheme. This finding is somewhat different from previous results where ANN (Fox & Shaw, 1994), LTC (Fuhr & Pfeifer, 1994) weighting schemes on query terms, LNC.LTC (Buckley et al., 1994) and LNC.LTN (Knaus & Shauble, 1994) combined weighting schemes on document terms and query terms showed the best results. On the other hand, our findings were quite similar to the result presented by Savoy (2003), where the ATN.NTC showed the best performance among the existing weighting schemes in SMART for English monolingual runs.

5.3 Evaluations in CLIR

Bilingual translations were extracted from the collection of news articles using the simple translation model and the two-stage translation model. A fixed number p (set to five) of top- ranked translation alternatives was retained for evaluations in CLIR. Results and performances on the monolingual run as well as on the bilingual runs using the two-stage corpus-based translation model and the linear combination to bilingual dictionaries are illustrated in Table 1. Evaluations are based on the average precision, differences in term of average precision of the monolingual counterpart and the improvement over the monolingual counterpart.

Retrieval methods are represented by the monolingual retrieval *Mono*, dictionary-based translation *DT*, the simple corpus-based translation model *SCT*, the bidirectional corpus-based translation model *BCT,* the two-stage corpus-based translation model *TCT*. Linear combinations were represented by *SCT+DT* for the combined simple corpus-based translation and bilingual dictionaries, *BCT+DT* for the combined bidirectional corpus-based translation and bilingual dictionaries, *TCT+DT* for the combined two-stage corpus-based translation and bilingual dictionaries. Our interest

Table 1. Evaluations on the proposed translation models using ATN.NTC weighting schemes of SMART retrieval system

Average Precision, % Monolingual, and % Improvement (*P=5*)							
Mono	*DT*	*SCT*	*BCT*	*TCT*	*SCT+DT*	*BCT+DT*	*TCT+DT*
0.3368 (100%)	0.2279 (67.66%)	0.1417 (42.07%)	0.1801 (53.47%)	**0.2008 (59.62%)**	0.2366 (70.25%)	0.2721 (80.79%)	**0.2987 (88.69%)**

in the present research is related to the evaluation of the proposed two-stage translation model *TCT* and the combination to bilingual dictionaries *TCT+DT* over the other retrieval methods. As illustrated in Table1, the bi-directional corpus-based translation model *BCT* showed a better improvement in terms of average precision compared to the simple corpus-based translation model *SCT* with +27.1%.

The two-stage corpus-based translation model *TCT* showed better performance in terms of average precision with +41.7% and +11.5% compared to *SCT* and to *BCT*, respectively. Linear combination of simple or comparable corpora and bilingual dictionaries showed better performances in terms of average precision compared to the models stand-alone. *SCT+DT* showed 70.25% of the monolingual counterpart *Mono*, +3.82% compared to the dictionary-based translation *DT* and +66.97% compared to the simple corpus-based translation *SCT*, in the case of ATN.NTC weighting scheme. *BCT+DT* showed better improvement with 80.79% of the monolingual counterpart *Mono*, +19.4% of the dictionary-based translation *DT*, +51.08% of *BCT*, and +15% of the combined *SCT+DT*. The proposed hybrid combination *TCT+DT* of the two-stage corpus-based translation model and bilingual dictionaries showed the best performance with 88.69% of the monolingual retrieval, in the case of ATN.NTC weighting scheme. *TCT+DT* showed an improvement of +9.7% of the combined *BCT+DT* and +26.24 of the combined *SCT+DT*. Furthermore, the different improvements in term of average precision were noticed through all weighting schemes of SMART retrieval system, with ATN.NTC showing the best results for all the retrieval methods involved in the present study.

Thus, key techniques used in the proposed two-stage corpus-based translation model for bilingual terminology acquisition from comparable corpora, can be summarized as follows:

The acquisition of bilingual terminology from bi-directional comparable corpora yields a significantly better result than using the simple model. The bi-directional corpus-based translation model is considered as one kind of symmetric probabilistic model that provides a disambiguation of the extracted translation alternatives and helps improve the accuracy of translation extraction. The bi-directional approach is more effective than the simple approach in a way that it includes a disambiguation process for the extracted translation alternatives. Evaluations in CLIR showed an improvement of 27.1% in terms of mean average precision for the bi-directional corpus-based translation model of the simple corpus-based translation model (the main average precision goes from 0.1417 to 0.1801 for the ATN.NTC weighting scheme).

The approach based on bi-directional comparable corpora largely affected the translation because related words could be added as translation alternatives or expansion terms.

Linguistics-based pruning technique has allowed a great improvement in the effectiveness of CLIR. Therefore, morphological knowledge such as part-of-speech could provide a valuable resource in filtering and pruning the translation candidates.

Combining different translation models yields a significantly better result than using each model by itself. Translation models based on comparable corpora and bilingual dictionaries have completed each other and their linear combination has provided a valuable resource for query translation/expansion in CLIR and has allowed an improvement in the effectiveness of information retrieval.

6 Conclusion

In the present paper, we investigated the approach of extracting bilingual terminology from comparable corpora in order to enhance CLIR, especially in the disambiguation and query expansion processes, and possibly enrich existing bilingual lexicons. We proposed a two-stage corpus-based translation model consisting of bi-directional extraction of bilingual terminology and linguistic-based pruning. Among the drawbacks of the proposed translation process is the introduction of many noisy terms or wrongly translated terms; however, most of those terms could be considered as efficient for the query expansion in CLIR but not for the translation.

Combination of two-stage corpus-based translation model and bilingual dictionaries yields to better translations and an effectiveness of information retrieval could be achieved across Japanese and English languages.

Further extensions include an integration of a transliteration model in the hybrid combination, especially for loanwords and foreign words of Japanese language. Second possible extension is the decomposition of the large-scale corpora into comparable pieces, instead of taking the whole corpus as a single piece, could be investigated in the future. Third, translation on a word-by-word basis is not applicable to all compounds and technical terms, especially when dealing with Japanese language. Thus, the problem of multiword phrases and compounds should be solved. Evaluations using other combinations and more efficient weighting schemes that are not included in SMART retrieval system such as OKAPI, which showed great success in information retrieval, are among the future subjects of our research on CLIR.

References

1. Buckley, C., Allan, J., Salton, G.: Automatic Routing and Ad-hoc Retrieval using SMART. In: Proceedings of the Second Text Retrieval Conference TREC-2, pp. 45–56 (1994)
2. Cancedda, N., Dejean, H., Gaussier, E., Renders, J.M., Vinokourov, A.: Report on CLEF-2003 Experiments : Two ways of Extracting Multilingual Resources from Corpora. In: Proceedings of LEF 2003 Evaluation Campaign, Norway, Trondheim, August 21-22 (2003)
3. Dagan, I., Itai, I.: Word Sense Disambiguation using a Second Language Monolingual Corpus. Computational Linguistics 20(4), 563–596 (1994)
4. Dejean, H., Gaussier, E., Sadat, F.: An Approach based on Multilingual Thesauri and Model Combination for Bilingual Lexicon Extraction. In: Proceedings of COLING 2002, Taiwan, pp. 218–224 (2002)
5. Diab, M., Finch, S.: A Statistical Word-Level Translation Model for Comparable Corpora. In: Proceedings of the Conference on Content-based Multimedia Information Access RIAO (2000)
6. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19(1), 61–74 (1993)
7. EDR. Japan Electronic Dictionary Research Institute, Ltd. EDR electronic dictionary version 1.5 technical guide. Technical report TR2-007. Japan Electronic Dictionary research Institute, Ltd. (1996)

8. Fox, A.E., Shaw, A.J.: Combination of Multiple Searches. In: Proceedings of the Second Text Retrieval Conference TREC-2, pp. 243–252 (1994)
9. Fuhr, N., Pfeifer, U., Bremkamp, C., Pollmann, M., Buckley, C.: Probabilistic learning Approaches for Indexing and Retrieval with the TREC-2 Collection. In: Proceedings of the Second Text Retrieval Conference TREC-2, pp. 67–74 (1994)
10. Fung, P.: A Statistical View of Bilingual Lexicon Extraction: From Parallel Corpora to Non-Parallel Corpora. In: Véronis, J. (ed.) Parallel Text Processing (2000)
11. Gaussier, E., Renders, J.M., Matveeva, I., Goutte, C., Dejean, H.: A Geometric View on Bilingual Lexicon Extraction from Comparable Corpora. In: Proceedings of ACL 2004, Barcelona, Spain, pp. 526–533 (2004)
12. Grefenstette, G.: The WWW as a Resource for Example-based MT Tasks. In: ASLIB 1999 Translating and the Computer 21 (1999)
13. Hedlund, T.: Compounds in dictionary-based cross-language information retrieval. Information Research 7(2) (January 2002)
14. Kaji, H.: Word Sense Acquisition from Bilingual Corpora. In: Proceedings of HLT-NAACL 2003, Edmonton, Canada, pp. 32–39 (2003)
15. Kando, N.: Overview of the Second NTCIR Workshop. In: Proceedings of the Second NTCIR Workshop on Research in Chinese and Japanese Text Retrieval and text Summarization, Tokyo (2001)
16. Klavens, J., Tzoukermann, E.: Combining Corpus and Machine-Readable Dictionary Data for Building Bilingual Lexicons. Machine Translation 10(3-4), 1–34 (1996)
17. Knaus, D., Shauble, P.: Effective and Efficient retrieval from large and Dynamic Document Collections. In: Proceedings of the Second Text Retrieval Conference TREC-2, pp. 163–170 (1994)
18. Knight, K., Graehl, J.: Machine Transliteration. Computational Linguistics 24(4) (1998)
19. Koehn, P., Knight, K.: Learning a Translation Lexicon from Monolingual Corpora. In: Proceedings of ACL 2002 Workshop on Unsupervised Lexical Acquisition (2002)
20. Matsumoto, Y., Kitauchi, A., Yamashita, T., Imaichi, O., Imamura, T.: Japanese morphological analysis system ChaSen manual. Technical Report NAIST-IS-TR97007, NAIST (1997)
21. Nakagawa, H.: Disambiguation of Lexical Translations Based on Bilingual Comparable Corpora. In: Proceedings of LREC 2000, Workshop of Terminology Resources and Computation WTRC 2000, pp. 33–38 (2000)
22. Nie, J.Y., Simard, M., Isabelle, P., Durand, R.: Cross-Language Information Retrieval based on parallel texts and automatic mining of parallel texts from the Web. In: Proceedings of the 22nd ACM SIGIR Conference, pp. 74–81 (1999)
23. Oard, D., Diekema, A.: Cross-Language Information Retrieval. In: Annual Review of Information Science and Technology (ARIST), vol. 33, pp. 223–256 (1998)
24. Peters, C., Picchi, E.: Capturing the Comparable: A System for Querying Comparable Text Corpora. In: Proceedings of the Third International Conference on Statistical Analysis of Textual Data, pp. 255–262 (1995)
25. Pirkola, A., Hedlund, T., Keskustalo, H., Järvelin, K.: Dictionary-based cross-language information retrieval: problems, methods, and research findings. Information Retrieval 4(3/4), 209–230 (2001)
26. Rapp, R.: Automatic Identification of Word Translations from Unrelated English and German Corpora. In: Proceedings of European Chapter of the Association for Computational Linguistics, EACL (1999)

27. Renders, J.M., Dejean, H., Gaussier, E.: Assessing Automatically Extracted Bilingual Lexicons for CLIR in Vertical Domains: XRCE Participation in the GIRT Track of CLEF 2002. In: Peters, C., Braschler, M., Gonzalo, J. (eds.) CLEF 2002. LNCS, vol. 2785, Springer, Heidelberg (2003)
28. Sadat, F., Maeda, A., Yoshikawa, M., Uemura, S.: Exploiting and Combining Multiple Resources for Query Expansion in Cross-Language Information Retrieval. IPSJ Transactions of Databases 43(SIG 9) (TOD 15), 39–54 (2002)
29. Sadat, F., Yoshikawa, M., Uemura, S.: Enhancing Cross-language Information Retrieval by an Automatic Acquisition of Bilingual Terminology from Comparable Corpora. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2003, Toronto, Canada (2003)
30. Sadat, F., Yoshikawa, M., Uemura, S.: Learning Bilingual Translations from Comparable Corpora to Cross-Language Information Retrieval: Hybrid Statistics-based and Linguistics-based Approach. In: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages IRAL 2003, Sapporo, Japan,
31. Sadat, F., Yoshikawa, M., Uemura, S.: Bilingual Terminology Acquisition from Comparable Corpora and Phrasal Translation to Cross-Language Information Retrieval. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, ACL 2003, Sapporo, Japan (2003)
32. Sadat, F.: Knowledge Acquisition from Collections of News Articles to Cross-language Information Retrieval. In: Proceedings of RIAO 2004 Conference (Recherche d'Information Assisté par Ordinateur), Avignon, France, April 26-28, pp. 504–513 (2004)
33. Salton, G.: The SMART Retrieval System, Experiments in Automatic Documents Processing. Prentice-Hall, Inc., Englewood Cliffs (1971)
34. Salton, G., McGill, J.: Introduction to Modern Information Retrieval. Mc Graw-Hill, New York (1983)
35. Savoy, J.: Cross-Language Information Retrieval: Experiments based on CLEF 2000 Corpora. Information Processing and Management 39(1), 75–115 (2003)
36. Sekine, S.: OAK System– Manual. New York University (2001)
37. Shahzad, I., Ohtake, K., Masuyama, S., Yamamoto, K.: Identifying Translations of Compound Using Non-aligned Corpora. In: Proceedings of the Workshop MAL, pp. 108–113 (1999)
38. Tanaka, K., Iwasaki, H.: Extraction of Lexical Translations from Non-Aligned Corpora. In: Proceedings of COLING (1996)
39. Utsuro, U., Horiuchi, T., Chiba, Y., Hamamoto, T.: Semi-automatic Compilation of Bilingual Lexicon Entries from Cross-Lingually Relevant News Articles on WWW News Sites. In: Proceedings of the Association for Machine Translation in the Americas (AMTA 2002), pp. 165–176 (2002)
40. Utsuro, T., Horiuchi, T., Hamamoto, T., Hino, K., Nakayama, T.: Effect of Cross-Language IR in Bilingual Lexicon Acquisition from Comparable Corpora. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003), pp. 355–362 (2003)

Semi-automatic Endogenous Enrichment of Collaboratively Constructed Lexical Resources: Piggybacking onto Wiktionary

Franck Sajous[1], Emmanuel Navarro[2], Bruno Gaume[1], Laurent Prévot[3], and Yannick Chudy[1]

[1] CLLE-ERSS, CNRS & Université de Toulouse
[2] IRIT, CNRS & Université de Toulouse
[3] LPL, CNRS & Université de Provence

Abstract. The lack of large-scale, freely available and durable lexical resources, and the consequences for NLP, is widely acknowledged but the attempts to cope with usual bottlenecks preventing their development often result in dead-ends. This article introduces a language-independent, semi-automatic and endogenous method for enriching lexical resources, based on collaborative editing and random walks through existing lexical relationships, and shows how this approach enables us to overcome recurrent impediments. It compares the impact of using different data sources and similarity measures on the task of improving synonymy networks. Finally, it defines an architecture for applying the presented method to Wiktionary and explains how it has been implemented.

Keywords: Collaboratively Constructed Lexical Resources, Endogenous Enrichment, Crowdsourcing, Wiktionary, Random Walks.

1 Introduction

While emerging processes of creation and diffusion keep increasing the production of digital documents, the tools to process them still suffer from a lack of acceptable linguistic resources for most languages. *"We desperately need linguistic resources!"* is claimed in [1], after arguing that it is not realistic to assume that large-scale resources can all be developed by a single institute or a small group of people, and concluding that a collaborative effort is needed, and that sharing resources is crucial. In this paper, we propose a new method for developing lexical resources which could meet these needs and we apply it to Wiktionary,[1] the free online dictionary. The system we describe automatically computes semantic relations, namely synonyms, to be added or not to a lexical network, after being validated or invalidated by contributors. In Section 2, we take inventory of the usual approaches and point out the impediments that hinder the success of such processes. We then investigate new trends which could help overcome this shortcoming. We outline in Section 3 the key points of our method, based on a

[1] http://www.wiktionary.org

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 332–344, 2010.

semi-automatic endogenous enrichment process. We explain in Section 4 how we compute the candidate relations by random walks over various graphs and using several measures that we evaluate, regarding our specific purpose. We present the architecture built to carry out the whole enrichment/validation system in Section 5 and we describe possible future extensions of out method in Section 6.

2 Lexical Resource Building

2.1 Context

Princeton WordNet [2] is probably the only successful large-scale project among lexical resource building attempts which is widely used. The subsequent projects EuroWordNet [3] and BalkaNet [4] were less ambitious in terms of coverage. Moreover, these resources froze as soon as the projects ended while Princeton WordNet kept on evolving. EuroWordNet's weaknesses have been underlined in [5], and automatic methods to add missing lexical relations have been proposed. Existing resources have been used in [6] to build WOLF, a *free* French WordNet. Pattern-based approaches were first proposed in [7] to harvest semantic relations from corpora and refined in [8] by reducing the need for human supervision. All of the latter three automatic processes would require validation by experts to produce reliable results. However, the cost of this validation work makes it difficult to afford or results in resources that are not freely accessible. The problems of time, cost and availability are increasingly becoming a matter of concern: in corpus-linguistics, an AGILE-like method borrowed from Computer Science has been proposed in [9] to address the problem of simultaneously maximizing corpus size and annotations while minimizing the time and cost involved in the creation of corpora. To tackle the availability issue and build free corpora, a method relying on metadata to automatically detect copylefted web pages is described in [10]. In the domain of lexical resource building, methods relying on *crowdsourcing* may help overcome recurrent bottlenecks.

2.2 Collaboratively Constructed Resources (CCR)

It has been claimed in [11] that the accuracy of Wikipedia comes close to Britanica, who criticized the criteria of the evaluation [12]. A more moderate study [13] has shown in a task measuring the semantic relatedness of words that resources based on the *"wisdom of crowds"* are not superior to resources based on the *"wisdom of linguists"*, but that CCRs are strongly competitive. It has also been demonstrated that "crowds" can outperform linguists in term of coverage.

Collaborative and social approaches to resource building do not rely only on colleagues or students but on random people, who do not share the NLP researchers' interest for linguistic resource building. Therefore, building sophisticated and costly infrastructures that are empty shells waiting to be filled presents the risk of being platforms that no one would visit. Indeed, in the current web landscape, competition for visitors is difficult and empty shells, as promising as

they can be, are not attracting many people. Any infrastructure that underestimates and does not answer this "attractiveness" issue is doomed to fail. However, there are at least two main tracks to follow in order to avoid this pitfall:

Gamers. Some language resource builders have been successful in designing simple web games in which many people come to play just for fun. For instance, the French serious game *"Jeux de Mots"*[2] [14] has been useful for collecting a great number of relations between words (mostly non-typed associative relations but also better defined lexico-semantic relations such as hypernymy, meronymy, etc.). However, setting up an interesting game for collecting any kind of linguistic information is not easily feasible. For instance, domain-specific resources might be harder to collect this way. Secondly, designing game-play that really works is a difficult task in itself and it is likely that many "game-elicited" resource initiatives will fail because of the game not being fun for random people.

Piggybackers. Only a few collaborative or social infrastructures are really successful. These resources and networks concentrate the majority of internet users. Merely being associated with one of these "success stories" affords the possibility of crowds of visitors. Wiktionary and Wikipedia are probably the best examples. The NLP community can offer some services to the users of these resources in order to take advantage of their huge amounts of visitors and contributors. Significant steps towards such an architecture have been made in [15,16]. Generalizing this approach to social networks, while adding a gaming dimension is also possible and constitutes an interesting avenue to be explored. Moreover, simply adding plugins to existing solid and popular infrastructures requires much less effort and technical skill than setting-up the whole platform (though lots of technical difficulties occur to comply with and plug into these infrastructures).

3 Outline of Proposal for a New Approach

Taking into account the observations made in Section 2 and considering the benefits of using CCRs, we propose a method for enhancing lexical resources that is reasonable in terms of time and cost, based on: (i) piggybacking onto Wiktionary, (ii) computing similarity measures grounded on random walks through the graphs extracted from its lexical networks (Sections 4.2 and 4.3) and (iii) giving an easy way for users to validate the candidate relations that we suggest.

3.1 Wiktionary

Wiktionary is a free multilingual collaborative dictionary including definitions, semantic relations and translations (a detailed presentation can be found in [15,16]). Its intrinsic features fulfill some of our needs: it is publicly available, its growth is fast and continuous and, as its content is based on crowdsourcing, the

[2] See http://www.lirmm.fr/jeuxdemots/jdm-accueil.php

"reasonable cost" constraint turns euphemistic. However, what is the quality of resources constructed by "naive speakers" as compared to those built by skilled professional lexicographers? A recent study [17] evaluated three German resources designed in different manners: expert-built (GermaNet), semi-controlled (OpenThesaurus) and collaboratively edited (German Wiktionary). This comparison demonstrated that all resources have a similar topology[3] and lexical coverage, but different density of semantic relations: for instance, Wiktionary has fewer hypernyms/hyponyms than GermanNet, but clearly outperforms both other resources in term of antonymy relations. Table 1 gives the number of common nouns, verbs, adjectives and (undirected) synonymy and translation links for the French and English Wiktionaries in 2008 and 2010. These figures relate to all lexemes found—conversely, in [16], only the lexemes connected by synonymy links have been counted. Translation and synonymy links have been counted after the graphs have been symmetrized (i.e. two-way links are counted once).

Table 1. Growth of French and English Wiktionaries from year 2008 to 2010

		2008			2010		
		Nouns	**Verbs**	**Adj.**	**Nouns**	**Verbs**	**Adj.**
FR	Lexemes	38 973	6 968	11 787	106 068 (×2.7)	17 782 (×2.6)	41 725 (×3.5)
	Synonymy links	9 670	1 793	2 522	17 054 (×1.8)	3 158 (×1.8)	4 111 (×1.6)
	Translation links	106 061	43 319	25 066	153 060 (×1.4)	49 859 (×1.2)	32 949 (×1.3)
EN	Lexemes	65 078	10 453	17 340	196 790 (×3.0)	67 649 (×6.5)	48 930 (×2.8)
	Synonymy links	12 271	3 621	4 483	28 193 (×2.3)	8 602 (×2.4)	9 574 (×2.1)
	Translation links	172 158	37 405	34 338	277 453 (×1.6)	70 271 (×1.9)	54 789 (×1.6)

As we can see, the number of lexemes has seen a growth that makes Wiktionary, for these languages, comparable to commercial printed dictionaries in term of lexical coverage: the French *"Petit Robert"* includes 60 000 entries and the *"Longman Dictionary of Contemporary English"* features 50 000 entries. Moreover, all the resources that capture some aspect of linguistic knowledge can prove to be useful and interesting. So, traditional resources and collaborative resources should both continue to be developed, especially since, as mentioned by [19], their content does not overlap too much.

Regarding semantic relations, we have shown the sparseness of the synonymy networks extracted from Wiktionary in 2008 [16]. Synonymy relations grew at slower rate than lexeme coverage, which makes the 2010 graphs even more sparse. To help fill this gap, we present below an endogenous enrichment method.

3.2 Endogenous Enrichment

Our aim is to be able to propose, for an existing semantic lexical network, new relations that are potentially missing. To propose new pairs of words which may be synonymous, we compute a similarity measure between any two nodes (lexemes) of the network by applying random walks through already existing lexical relations. Details of the different data sources, graph modeling and measures we use are given in Section 4.

[3] Extracted graphs are small worlds with a heavy-tailed degree distribution: see [18].

As the potential new synonyms we compute are to be validated by contributors, and not automatically added to the initial resource, our purpose is: (i) to suggest candidates for the greatest number of lexemes and (ii) for a given lexeme, to propose a finite list of candidates including at least *"some"* relevant ones.

In our case, it is better to propose no candidate at all than irrelevant ones, and the system is not meant to suggest *all* relevant candidates: first, because a contributor won't check an endless list and secondly, our method is an iterative computation-suggestion-validation cycle. Thus, if a relevant candidate is not initially proposed, it may be the next in the list of suggestions, which may be shifted when a suggested candidate is chosen. So, as the relations added to the network will change its structure, and as the computation of candidates will be reprocessed regularly (after the release of a new dump in the case of Wiktionary), this relevant candidate may be proposed after some iterations. Thus, recall will increase with successive iterations and we focus therefore more on precision.

3.3 Validation

The candidates that we compute are suggested to the contributors via an interface described in Section 5. If a contributor validates a suggestion, the relation is added to Wiktionary. No cross-validation system, in which a relation would be added only if several contributors validate it, has been designed: to keep close to the wiki principle, we did not add any additional regulation,[4] but as we ease the addition of synonyms, we fairly give an easy way to remove them too.

4 Similarity Elicitation

This section presents the methods used to compute, from existing lexical networks, new synonymy relations to be added. We rely on different kinds of data and similarity measures and compare the results obtained by evaluating them against expert-built gold standards.

4.1 Data

Networks have been extracted from English and French Wiktionaries for nouns, verbs and adjectives, thus splitting the global structure of the dictionaries into mono-part of speech subparts. Given a language version of Wiktionary, we consider only the article sections dedicated to entries in the language of interest, e.g. the English lexemes of the English Wiktionary. From these sections, we extract the existing synonymy and translation links, as well as the glosses.

4.2 Bipartite Graphs Model

In order to homogenize and simplify the description of the experiments, each type of data we used will be modelled as a *undirected bipartite graph* $G = (V \cup V', E)$

[4] For some insights into the autoregulation of the Wikiprojects ecosystem, see [20].

where the set of vertices V will always denote the lexemes of the language and part of speech of interest, whereas another set of vertices V' will vary depending on the sources of data. The set of edges E is such that $E \subseteq (V \times V') \cup (V' \times V)$ and models the relations between the lexemes of V and of V'.

- **Translation graph $\mathbf{G_{Wt}} = (\mathbf{V} \cup \mathbf{V_{Wt}}, \mathbf{E_{Wt}})$.** Here, $V' = V_{Wt}$ is the set of the lexemes in all languages but the one of interest. E_{Wt} is the set of translation links: there is an undirected edge between $v \in V$ and $t \in V_{Wt}$ if t is found as a translation of v.[5]
- **Synonymy graph $\mathbf{G_{Ws}} = (\mathbf{V} \cup \mathbf{V_{Ws}}, \mathbf{E_{Ws}})$.** Here, $V' = V_{Ws}$ is a copy of V. There is an undirected edge between $v \in V$ and $u \in V_{Ws}$ either if $v = u$ or if u (or v) is indicated as a synonym in v (or u) entry. This bipartite graph model of the synonymy network may look unusual, however: (i) it permits us to have a unique bipartite graph model, (ii) for the random walk algorithms presented below, this model is equivalent to a classic unipartite synonymy network.
- **Glosses graph $\mathbf{G_{Wg}} = (\mathbf{V} \cup \mathbf{V_{Wg}}, \mathbf{E_{Wg}})$.** Here, $V' = V_{Wg}$ corresponds to the set of all lemmatized lexemes found in the glosses of all entries. There is an undirected edge between $v \in V$ and $g \in V_{Wg}$ if g is used in one of the definitions of v. For a given lexeme, its glosses have been concatenated, lemmatized and tagged with Treetagger,[6] and stopwords have been removed.
- **Syns+Trans graph $\mathbf{G_{Ws+t}} = (\mathbf{V} \cup \mathbf{V_{Ws}} \cup \mathbf{V_{wt}}, \mathbf{E_{Wst}} = \mathbf{E_{Ws}} \cup \mathbf{E_{Wt}})$.** Here, $V' = V_{Ws} \cup V_{wt}$ includes a copy of the set of lexemes V and their translations. There is an undirected edge between $v \in V$ and $v' \in V_{Ws} \cup V_{Wt}$ if v' is either a synonym or a translation of v.

4.3 Random Walk-Based Similarity Computation

To propose new synonymy relations, we compute the similarity between any possible pair of lexemes (the vertices from the graphs described in the previous section). The intent is to propose as candidates the pairs with the highest scores (which are not already known as synonyms in Wiktionary). We test various similarity measures, all based on—short—fixed length random walks. Such approaches for measuring the "topological resemblance" in graphs are introduced in [18,21]. This kind of methods is applied to lexical networks in [22] to compute semantic relatedness. We consider a walker wandering at random in the *undirected bipartite graph* $G = (V \cup V', E)$, starting from a given vertex v. At each step, the probability for the walker to move from nodes i to j is given by the cell (i, j) of the transition matrix P, defined as follow:

$$[P]_{ij} = \begin{cases} \frac{1}{d(i)} & \text{if} \quad (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where $d(i)$ is the degree of incidence (number of neighbours) of vertex i. Thus, starting from v, the walker's position after t steps is given by the distribution of

[5] As we parse only the dump of the language of interest, we find the *oriented* link $v \rightarrow t$ (t as a translation of v in v's article) and symmetrize it into $v \leftrightarrow t$. Having a more subtle model (with oriented edges) requires parsing all dumps of all languages.

[6] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

probabilities $X_t(v) = \delta_v P^t$, where δ_v is a row vector of dimension $|V \cup V'|$ with 0 anywhere except 1 for the column corresponding to vertex v. We note $X_t(v, u)$ the value of the coordinate u of this vector, which denotes as aforementioned the probability of reaching u afer t steps, starting from v. This is the first measure[7] (called *simple*) we use ; other measures are based on this one:

$$\text{simple}(v, u) = X_t(v, u) \tag{2}$$

$$\text{avg}(v, u) = \frac{X_t(v, u) + X_t(u, v)}{2} \tag{3}$$

$$\cos(v, u) = \frac{\sum_{w \in V} X_t(v, w) X_t(u, w)}{\sqrt{\sum_{w \in V} X_t(v, w)^2} \sqrt{\sum_{w \in V} X_t(u, w)^2}} \tag{4}$$

$$\text{dot}(v, u) = \sum_{w \in V} X_t(v, w) X_t(u, w) \tag{5}$$

$$\text{ZKL}_\gamma(v, u) = \sum_{w \in V} X_t(v, w) \begin{cases} \log(\frac{X_t(v,w)}{X_t(u,w)}) \text{ if } X_t(u, w) \neq 0 \\ \gamma \text{ otherwise} \end{cases} \tag{6}$$

"cos" and "dot" are respectively the classical cosine and scalar product. "ZKL_γ" is a variant of the Kullback-Leibler divergence introduced in [22].

Let $C(v, G, t, \text{sim})$ be the ordered list of candidates computed on graph G with the similarity measure *"sim"* and a random walk of length t, starting from v:

$$C(v, G, t, \text{sim}) = [u_1, u_2, \ldots, u_n] \quad \text{with} \quad \begin{cases} \forall i, \text{sim}(v, u_i) \geq \text{sim}(v, u_{i+1}) \\ \forall i, \text{sim}(v, u_i) > 0 \\ \forall i, (v, u_i) \notin E_{Ws} \end{cases} \tag{7}$$

where E_{Ws} is the set of existing synonymy links in Wiktionary. The experiments below consist in evaluating the relevancy of $C(v)$ when G and *sim* vary. $t = 2$ will remain constant.[8]

4.4 Evaluation Method

In view of our application (cf. Section 5.2) and given the criteria defined in Section 3.2, for each lexeme, we consider that a *suggested list* of candidates is *acceptable* if it includes at least one relevant candidate. Indeed, a user can contribute provided that at least one good candidate occurs in the suggested list. Thus, the evaluation will broadly consist in counting for how many lexemes the system computes a suggested list with at least one relevant candidates.

Let $G_{GS} = (V_{GS}, E_{FS})$ be a gold standard synonymy network, where V_{GS} is a set of lexemes, and $E_{GS} \subseteq V_{GS} \times V_{GS}$ a set of synonymy links. We evaluate below the acceptability of the suggested lists made to enhance the deficient resource

[7] All these measures are not strictly speaking *similarity* ; indeed, "simple" and "zkl10" are not symmetric.

[8] t has to be even and preliminary experiments have shown that the best results are obtained with 2 or 4. $t = 2$ gives similar results and is less complex.

against the gold standard's relations. We only evaluate the suggested lists for the lexemes that are "known" by the gold standard (i.e. $v \in V_{GS}$). Indeed, if a lexeme $v \in V$ does not belong to the gold standard (i.e. $v \notin V \cap V_{GS}$), we consider that it is a lexical coverage issue, so one cannot deem whether a relation (v, c) is correct or not.[9] For the same reason, for each lexeme v, we remove from $C(v)$ the candidates absent from the gold standard. Finally we limit the maximum number of candidates to $k \leq 5$. For each lexeme $v \in V \cap V_{GS}$, we note $\Gamma_k(v)$ the "evaluable" suggested list of candidates:

$$\Gamma_k(v) = [c_1, c_2, \ldots, c_{k'}] \quad \text{with} \quad \begin{cases} k' \leq k \\ \forall i,\ c_i \in C(v) \cap V_{GS} \\ \forall i, \mathrm{sim}(v, c_i) \geq \mathrm{sim}(v, c_{i+1}) \end{cases} \tag{8}$$

Please note that $\Gamma_k(v)$ contains a maximum of k candidates (but it may be smaller or even empty). Note also that $\Gamma_k(v)$ depends on the gold standard. We note $\Gamma_k^+(v)$ the set of correct candidates within $\Gamma_k(v)$:

$$\Gamma_k^+(v) = \left\{ c^+ \in \Gamma_k(v) / (v, c^+) \in E_{GS} \right\} \tag{9}$$

We define the set N_k of lexemes having at least one candidate being proposed and the set N_k^+ of lexemes for which at least one *correct* candidate is proposed:

$$N_k = \left\{ v \in V \cap V_{GS} / \Gamma_k(v) \neq \varnothing \right\}, N_k^+ = \left\{ v \in V \cap V_{GS} / \Gamma_k^+(v) \neq \varnothing \right\} \tag{10}$$

To compare the efficiency of different data sources used to compute the candidates, we measure P_k, the ratio between the *acceptable* suggested lists and the lexemes for which suggestions are done, and R_k, the ratio between the number of suggested lists and the number of evaluable target lexemes:

$$P_k = \frac{|N_k^+|}{|N_k|}, R_k = \frac{|N_k|}{|V_{GS} \cap V|} \tag{11}$$

Although P_k and R_k are not precision and recall measures, they intuitively refer to the same notions and we adopt below—abusively—this terminology.

4.5 Results

Gold Standards: We used Princeton WordNet to evaluate the candidates for English and DicoSyn[10] for French. The extraction of the synonymy networks from these resources reproduces what has been done in [16].

Similarity measures: Applying the different similarity measures presented in Section 4.3 shows that all give pretty similar results. As an example, the results obtained for the intersection of the gold standards and the English and French Wiktionaries' nouns and verbs are reported in Table 2. The *simple* measure being as efficient as the others and having far less complexity, further experiments have therefore been done using this measure.

[9] v may be a neologism or a domain-specific word. Less often, it may be misspelling. Any relation (v, c) should therefore not be counted as false (or true).

[10] Dicosyn is a compilation of synonym relations extracted from seven dictionaries produced at ATILF and corrected at CRISCO units.

Table 2. P_5 precision comparison for different data sources and measures

	Synonyms				Translations				Syn. + Trans.			
	EN		FR		EN		FR		EN		FR	
	V	N	V	N	V	N	V	N	V	N	V	N
simple	41.4	32.4	58.6	47.3	51.4	37.8	**78.7**	58.3	51.9	39.0	**74.6**	**55.3**
avg	42.5	33.5	58.2	46.8	50.5	38.0	**78.7**	58.3	51.1	39.3	74.0	55.1
cos	**43.4**	**34.6**	**60.2**	47.9	51.8	38.5	78.3	58.6	51.3	39.4	73.1	54.2
dot	42.0	34.0	59.7	46.7	**52.3**	**38.7**	78.2	58.7	**52.4**	39.7	73.6	54.8
ZKL_{10}	43.2	34.0	60.1	**48.2**	51.8	38.6	**78.7**	**58.8**	51.9	**39.8**	74.0	54.5

Data sources: As we can see in Table 3, better results are obtained for French than for English. This can be partly explained by the slightly lower density of the English networks (cf. Table 1) but is mainly due to the difference between the gold standards used: networks extracted from WordNet are more sparse than the ones extracted from Dicosyn (see [16]). Morever, Table 4 shows that some candidates rejected by the gold standards do not look unreasonable, which makes it hard to draw definitive conclusions. Nevertheless, despite a—potentially—severe evaluation, results look acceptable enough in view of our application.

The translations graph provides better precision than synonymy graphs. This result was expected, as in Wiktionary, lexemes have more translation links than synonyms. Moreover, translations are often distributed over several languages, which is more reliable than having a lot of translations into a given language.

The glosses graph's worse precision and higher recall was expected too: almost all lexemes have glosses, but information is less specific, and we did not try any tricky edge weighting. Combining synonyms and translations enables a better recall than with separated graphs and a similar precision for English. For French, it leads to a loss of precision compared to the "translations only" graph.

Table 3. Impact of different data sources on the *simple* similarity measure

					Synonyms		**Translations**		**Syn.+Trans.**		**Glosses**	
		$\lvert V\rvert$	$\lvert V_{GS}\rvert$	$\lvert V \cap V_{GS}\rvert$	P_5	R_5	P_5	R_5	P_5	R_5	P_5	R_5
EN	Adj.	48930	21479	13742	46.3	24.9	53.5	23.4	53.7	34.6	26.1	98.3
	Nouns	196790	117798	43236	32.4	17.1	37.8	24.9	39.0	32.4	14.9	98.9
	Verbs	67649	11529	8890	41.4	33.2	51.4	43.5	51.9	53.8	27.0	99.9
FR	Adj.	41725	9452	3958	61.2	24.9	76.1	19.8	69.6	34.2	32.2	96.1
	Nouns	106068	29372	16084	47.3	23.2	58.3	22.2	55.3	35.4	20.7	99.4
	Verbs	17782	9147	4037	58.6	22.3	78.7	36.8	74.6	45.8	41.1	99.4

Table 4. Example of propositions for nouns evaluated against gold standards (GS)

	in GS	**Propositions**
EN	Yes	*<imprisonment: captivity>, <harmony: peace>, <filth: dirt>, <antipasto: starter>, <load: burden>, <possessive: genitive>*
	No	*<rebirth: renewal>, <fool: idiot, dummy>, <cheating: fraud>, <bypass: circumvention>, <dissimilarity: variance>, <pro: benefit>*
FR	Yes	*<ouvrage, travail>, <renom: gloire>, <emploi: fonction>, <drapeau: pavillon>, <rythme: cadence>, <roulotte: caravane>, <chinois: tamis>*
	No	*<drogue: psychotrope>, <fantassin: bidasse>, <force: poigne>, <salade: bobard>, <W.C.: chiotte>, <us: tradition>, <bisque: soupe>*

5 Implementation: The WISIGOTH Architecture

In order to carry out our enrichment method, we designed an architecture called WISIGOTH[11] composed of a set of modules depicted in Fig. 1.

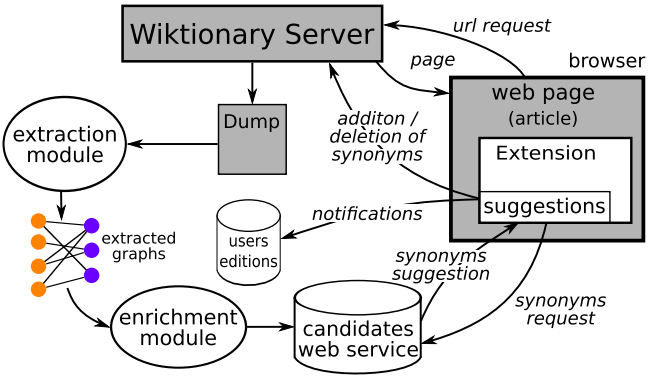


Fig. 1. The WISIGOTH architecture

5.1 Computation of Candidates

The first part of the architecture is made of a processing pipeline which, from a Wiktionary dump,[12] builds the graphs introduced in Section 4.2 and computes the candidate relations by applying the method described in Section 4.3. This processing pipeline can be triggered each time a new dump is released or when a given threshold of edits has been registered.

5.2 Suggestion and Validation of Candidates

The interface we developed to suggest and validate or invalidate new relations materializes as a Firefox extension. Once installed, when a user browses the English or French Wiktionary, the interface sends a request to the candidates service which returns, for each known lexeme, a list of potential synonyms.

Suggestion and Editing: Next to each proposition appears a '+' sign which triggers the automatic addition of the candidate as a synonym to the Wiktionary server. As a contributor may want to add a synonym that has not yet been suggested, we provide a free text area too. Regardless of our enrichment method, it enlarges the potential population of contributors, not restricting it to "wikicode-masters" acquainted with the underlying syntax. As explained is Section 3.3, a '−' sign is added to every synonym occuring in the page, which handles the deletion of this synonym.

[11] WIktionarieS Improvement by Graph-Oriented meTHods.

[12] Wiktionaries' dumps are available at: `http://download.wikipedia.org/`

Notification of editing: Thus far, wiktionaries dumps are released frequently. Nevertheless, to protect against irregular dumps which could result in a desynchronization between Wiktionary's current state and the lexical networks we extracted from it—and therefore, cause irrelevant suggestions—, the interface notifies our server the editing of synonyms. Thus, a remodelling of synonymy networks and a reprocessing of candidates may be done between two releases.

Storing these notifications will also later give us the opportunity to analyse which synonymy links look problematic (e.g. a series of additions and deletions) and how contributors behave.

6 Conclusion and Future Work

This paper has pointed out the problems usually encountered in the development of lexical resources. It has shown how CCRs help overcome these difficulties and, among them, how we can take advantage of Wiktionary's infrastructure and content. Nevertheless, "crowds" are more prone to add new words than to provide semantic relations. To encourage them, we have designed a tool to assist collaborative editing by suggesting new synonyms to be validated. We took the opportunity to compare the impacts of using different data sources and similarity measures. The choice of the measure does not much affect the results whereas combining data sources permits us to gain precision or recall, depending on the language. Adding glosses to the *"Syn+Trad"* graph presented and working on the weighting of the graphs' edges should bring even better results.

Grounded on the topology of the graphs extracted from the lexical networks, this system is language-independent and, moreover, may be applied to other resources than Wiktionary, contrary to methods like [23] which exploit the structure of hyperlinks between pages and are therefore bound to this resource. It may help, for example, building WordNets that are still under construction, as the Chinese one [24]. Moreover, not relying on other external resources makes this method endogenous and may be applicable to enhance lexical resources for under-resourced languages. When external resources are available, for example stemming from distributional analysis over large corpora, an exogenous enrichment module can be coupled to our system and feed our edition interface.

A short-term extension of this work will be the proposition of new translations by leveraging the same kind of graph model and similarity measures. Linguistic observations should be done to characterize what other kinds of semantic relation (than synonymy) is captured by automatically computed relatedness.

Although we did not rely on a cross-validation system for adding synonyms, we think it could be useful to add a blacklist system to stop proposing a candidate judged as irrelevant by several contributors for a given target lexeme.

An interesting study would be the evaluation of the results of the endogenous enrichment process at different stages of Wiktionary's growth. This can be done by rebuilding the various past states of the lexical networks using the "historical dump" containing all articles revisions. Such a study may show when it is appropriate to apply our method: when we have enough material to start suggesting

new relations and when no more relevant relation is to be proposed and should be stopped.

Resources: the Firefox extension presented in this paper and the structured data extracted from Wiktionary's dumps are publicly available at: http://redac.univ-tlse2.fr/wisigoth/

References

1. Sekine, S.: We desperately need linguistic resources! –based on the users' point of view. In: FLaReNet Forum 2010, Barcelona, Spain (2010)
2. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
3. Vossen, P. (ed.): EuroWordNet: a Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers, Norwell (1998)
4. Tufis, D.: Balkanet Design and Development of a Multilingual Balkan Wordnet. Romanian Journal of Information Science and Technology 7 (2000)
5. Jacquin, C., Desmontils, E., Monceaux, L.: French EuroWordNet Lexical Database Improvements. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 12–22. Springer, Heidelberg (2007)
6. Sagot, B., Fišer, D.: Building a Free French Wordnet from Multilingual Resources. In: Proceedings of OntoLex 2008, Marrakech (2008)
7. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of the 14th International Conference on Computational Linguistics (COLING), Nantes, pp. 539–545 (1992)
8. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In: Proceedings of the International Conference on Computational Linguistics, Sydney, pp. 113–120. ACL Press (2006)
9. Voormann, H., Gut, U.: Agile Corpus Creation. Corpus Linguistics and Lingustic Theory 4, 235–251 (2008)
10. Brunello, M.: The Creation of Free Linguistic Corpora from the Web. In: Proceedings of WAC5: 5th Workshop on Web As Corpus, San Sebastian, pp. 37–44 (2009)
11. Giles, J.: Internet Encyclopaedias Go Head to Head. Nature 438, 900–901 (2005)
12. Encyclopaedia Britannica: Fatally Flawed: Refuting the Recent Study on Encyclopedic Accuracy by the Journal Nature (2006)
13. Zesch, T., Gurevych, I.: Wisdom of Crowds versus Wisdom of Linguists – Measuring the Semantic Relatedness of Words. Journal of Natural Language Engineering 16, 25–59 (2010)
14. Lafourcade, M.: Making People Play for Lexical Acquisition with the JeuxDeMots prototype. In: SNLP 2007: 7th International Symposium on Natural Language Processing, Pattaya, Thailand (2007)
15. Zesch, T., Müller, C., Gurevych, I.: Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: Proceedings of the Conference on Language Resources and Evaluation (LREC), Marrakech (2008)
16. Navarro, E., Sajous, F., Gaume, B., Prévot, L., Hsieh, S., Kuo, I., Magistry, P., Huang, C.R.: Wiktionary and NLP: Improving Synonymy Networks. In: Proceedings of the ACL-IJCNLP Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, Singapore, pp. 19–27 (2009)

17. Meyer, C.M., Gurevych, I.: Worth its Weight in Gold or Yet Another Resource – A Comparative Study of Wiktionary, OpenThesaurus and GermaNet. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 38–49. Springer, Heidelberg (2010)
18. Gaume, B., Venant, F., Victorri, B.: Hierarchy in Lexical Organization of Natural Language. In: Pumain, D. (ed.) Hierarchy in Natural and Social Sciences. Methodos series, pp. 121–143. Kluwer Academic Publishers, Dordrecht (2005)
19. Zesch, T.: What's the Difference? Comparing Expert-Built and Collaboratively-Built Lexical Semantic Resources. In: FLaReNet Forum 2010, Barcelona, Spain (2010)
20. Forte, A., Bruckman, A.: Scaling Consensus: Increasing Decentralization in Wikipedia Governance. In: Proceedings of the 41st Hawaii International Conference on System Sciences, Washington DC, p. 157. IEEE Computer Society, Los Alamitos (2008)
21. Gaume, B., Mathieu, F.: PageRank Induced Topology for Real-World Networks. Complex Systems (2008)
22. Hughes, T., Ramage, D.: Lexical Semantic Relatedness with Random Graph Walks. In: Proceedings of EMNLP-CoNLL, pp. 581–589 (2007)
23. Weale, T., Brew, C., Fosler-Lussier, E.: Using the Wiktionary Graph Structure for Synonym Detection. In: Proceedings of the ACL-IJCNLP Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, Singapore, pp. 28–31 (2009)
24. Huang, C.R., Chen, C.L., Weng, C.X., Lee, H.P., Chen, Y.X., Chen, K.J.: The Sinica Sense Management System: Design and Implementation. Computational Linguistics and Chinese Language Processing 10, 417–430 (2005)

Portable Extraction of Partially Structured Facts from the Web

Andrew Salway[1], Liadh Kelly[1], Inguna Skadiņa[2], and Gareth J.F. Jones[1]

[1] Centre for Digital Video Processing, School of Computing,
Dublin City University, Dublin 9, Ireland
{asalway,lkelly,gjones}@computing.dcu.ie
[2] Tilde, 75 Vienības Gatve, Rīga 1004, Latvia
inguna.skadina@tilde.lv

Abstract. A novel fact extraction task is defined to fill a gap between current information retrieval and information extraction technologies. It is shown that it is possible to extract useful partially structured facts about different kinds of entities in a broad domain, i.e. all kinds of places depicted in tourist images. Importantly the approach does not rely on existing linguistic resources (gazetteers, taggers, parsers, etc.) and it ported easily and cheaply between two rather different languages (English and Latvian). Previous fact extraction from the web has focused on the extraction of structured data, e.g. (Building-LocatedIn-Town). In contrast we extract richer and more interesting facts, such as a fact explaining why a building was built. Enough structure is maintained to facilitate subsequent processing of the information. For example, the partial structure enables straightforward template-based text generation. We report positive results for the correctness and interest of English and Latvian facts and for their utility in enhancing image captions.

Keywords: Fact extraction, multilingual, information retrieval, information extraction, web, image captioning.

1 Introduction

This paper proposes a novel fact extraction task which fills an important gap between current information retrieval (IR) and information extraction (IE) technologies in order to further exploit the vast quantities of multilingual information available on the web. Search engines retrieve relevant web pages across diverse domains and across languages, but the onus is on the user to read through and interpret the results. By contrast, IE systems provide structured facts and data from natural language texts which are amenable to further automated analysis, and multi-document summarization systems and question answering systems fuse information about an entity or topic of interest to reduce reading time. However, such systems are typically costly to port to new languages and the domains in which they work tend to be narrow and comprise only a small set of entity types and relations. We believe that there are emerging applications,

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 345–356, 2010.

such as automated image captioning and augmented reality, which would benefit from exploiting information on the web across broad domains and multiple languages, but which do not require fully structured information or the majority of all available information about an entity. For example, to automatically enhance an image caption we only require one interesting fact about the place in the image, with enough structure for the fact to be inserted appropriately into a text generation template. In sacrificing the requirements for full structure and comprehensive information about an entity, we expect to gain considerably in broad coverage of domains and ease of porting between languages.

We elaborate these points in Section 2 as we define the "Tell Me About..." task which is, roughly, to provide one or more of the most interesting facts about a given entity in a partially structured form that enables some further processing and re-use of the information. Section 3 discusses related work in the fields of IR and IE, with a focus on information extraction from the web, multi-document summarization and question answering. Section 4 presents a highly portable solution for extracting partially structured facts that exploits information redundancy on the web, i.e. the fact that the same information about an entity is available in many forms on the web. The crucial assumption is that at least one key fact about an entity will be expressed somewhere on the web in a simple form. This means that we work with a few simple linguistic structures and shallow language processing and so the solution ports easily between languages. We report positive results for the correctness and interest of facts in two rather different languages, English and Latvian (128 facts each judged by an investigator and five subjects). Latvian is a highly inflected language: nouns, adjectives, participles and verbs are all inflective and, because of this rich morphology, Latvian has quite free word ordering. The utility of the "Tell Me About..." task is demonstrated by enhancing the captions of tourist photographs using extracted facts for template-based text generation, with an evaluation of caption readability (90 image captions each judged by six subjects). In closing, Section 5 considers generalising our solution to other domains and applications.

2 The "Tell Me About..." Task

Let us elaborate on the details of this task, and the motivation for it, by considering one potential application - automatic image captioning. The number of digital images being archived in personal collections and shared in social image collections (such as `www.flickr.com` and `www.panoramio.com`) is increasing very rapidly. When users view images from these collections it is desirable to have information describing each image available in a caption. However, people taking pictures will often either not know sufficient details about the place depicted in the image to do this effectively or will not take the time to do this, so automated solutions are required. There is also a burgeoning interest in augmented reality whereby a camera screen on a mobile device is updated automatically with caption-like information about the place that the camera is pointed at. Digital image capture devices are increasingly incorporating location sensing via

GPS monitoring. This can be combined with other image metadata such as the date and time of capture and cross-referenced with geographic databases to generate simple descriptive captions for an image, e.g. of the form "North Bridge photographed in the afternoon" [1]. We see an opportunity to exploit the vast information content of the web in order to enhance such a caption with a key fact, e.g. to output something like "North Bridge, which was built to link the New Town with the Old Town, photographed in the afternoon".

Whilst we can be confident that information about many places is available in many languages on the web, the challenge is to identify the most interesting facts for a given entity. There is also the challenge of extracting information into partially structured facts that enable further processing and re-use of the information. In the image captioning scenario simply adding whole sentences from the web to an existing caption would have unpredictable results for caption readability. It could be that a long sentence contains information about more than one place, so we need to identify just the relevant part of the sentence. Also, if we want to insert information into an existing caption, i.e. into the middle of a sentence, then we need to know something about how it phrased. For the "Tell Me About..." task we specify that facts should have the form of a triple - (Entity, Cue, Text-Fragment), where 'Cue' is one of a fixed set of information cues (loosely akin to relations), and 'Text-Fragment' is a text fragment taken directly from a webpage, such that 'Cue Entity Text-Fragment' reads naturally as a sentence, e.g. (North Bridge, was built, to link the New Town with the Old Town). For template-based image captioning this means we can, for example, insert information in a subclause starting with "which" for cues such as 'was built', but removing "which" and the cue itself for cues like 'is'. The partial structure of the fact gives us control over text generation that we would not have if the fact was only a text fragment. However, because the right-hand side of the fact is a text fragment, and not another entity of fixed type (as it would be in a standard IE template), then the same cue can get quite different kinds of information, allowing for much richer facts when available, e.g. (Hadrian's Wall, was built, in AD 122-130 on the orders of the Emperor Hadrian), (Hadrian's Wall, was built, to keep out the marauding Scottish).

To summarise, the "Tell Me About..." task proposed here is as follows. Given the name of an entity, and a specified language, a list of facts about the entity should be returned in the form (Entity, Cue, Text-Fragment) sorted with interesting facts ranked higher. With regards to image captioning, it is important to note that the place depicted in a photo may be one of very many different kinds of entity (bridge, monument, beach, church, mountain, plaza, glacier, etc.). Furthermore, the most interesting aspect of one entity may not be the same as the most interesting aspect of another entity of the same type - one church has spectacular stained-glass windows, another is known for an historical event that happened there, a third offers amazing views from its tower. Finally, a caption for an image on a website may be required in many languages. For these reasons, as we discuss next, current IE approaches are not appropriate.

3 Related Work

Although we consider "Tell Me About..." to be distinct from other natural language processing tasks, it does clearly have similarity with established and well understood tasks within IR and IE. The idea of ranking facts could be seen as similar to the ranking of documents for IR [2], and, more specifically, the retrieval and ranking of passages [3]. Indeed, snippets returned by web search engines are the starting point in our approach to fact extraction, although by the end of the process the sorted facts are in a different order than the snippets ranked by the search engine. The extraction of partially-structured information makes our fact extraction look quite a lot like IE [4], but whilst we do specify a set of cues (similar to relations), we do not require the structuring of the right-hand side text fragment into a template (which would, for example, make relations between entities explicit). We have found that this makes it possible to pursue quite a generic approach to fact extraction across broad domains and multiple languages, whereas IE systems require non-trivial amounts of work to be adapted to different kinds of entities and languages. Question answering systems return facts, typically in response to factoid questions, with answers that are dates, locations, organizations, people, etc. [5]. However, for a given entity it is not possible to anticipate what, if any, factoid question will give the most interesting information. That said, our approach to fact extraction shares assumptions about the redundancy of information on the web with some question answering techniques, e.g. [6]. Multi-document summarization systems do something rather like the "Tell Me About..." task when they select a set of informative sentences about an entity, e.g. [7], but with a focus on more than just a few key facts, and the need to produce coherent text as output, such systems typically depend on quite extensive linguistic resources - at a minimum training corpora - that mitigate against porting easily between many languages.

Previous work on information extraction from the web, rather than from domain-specific collections of a single text type, has achieved impressive quantities of facts at high levels of precision, e.g. 1 million ranked facts with a prespecified relation at 75-98% Precision [8]. Under the rubric of "open information extraction", which discovers relations as well as facts, a precision of 88% has been reported [9]. In related work the TextRunner system extracted over 500 million tuples from 120 million web pages [10]. However, much of this previous work has focused on the extraction of wholly structured data to specify relations between two entities, e.g. facts of the form (City-CapitalOf-Country), (Person-BornIn-Year), or (Company-Acquired-Company). Whilst this effectively enables the storage, analysis and retrieval of millions of facts in relational databases, these relatively simple facts are unlikely to be interesting for applications such as image captioning. An online demonstration does suggest that the TextRunner system [11] can provide facts with unstructured right-hand sides but our impression is that low quality of information is the price for exceptionally broad coverage. Furthermore, with regards to portability between languages, the approaches described by [9] and [10] rely on a linguistic analysis of how relations are expressed in English, and on syntactic parsers. Although the approach in [8]

avoids syntactic analysis and parsing, it nevertheless works with text that has been part-of-speech tagged and draws on existing word distribution data. Taggers, parsers, and other linguistic resources are not available for many languages and so we have developed an approach that does not need them.

4 Our Approach to Fact Extraction

Here we present a first solution for the "Tell Me About..." task. We show how, given an entity (in this case any kind of place), we return a list of facts in the form (Entity, Cue, Text-Fragment), ranked according to a score which is intended to promote interesting and true facts. The approach is generic across a broad range of entities, and requires minimal effort to port between languages. It is based on two assumptions: (i) the same information is expressed in many ways across the web, so it is only necessary to look for it in a small number of relatively simple forms; and, (ii) overlaps between what is written on different web pages can be used to compute an interest/correctness score to rank facts.

4.1 Algorithm

Given an entity, steps I-IV generate a list of facts about it.

I. Get Snippets from Search Engine. A series of queries is made to a web search engine (we used Yahoo's BOSS API [12]). Each query takes the form <"Entity Cue">; the use of double quotes indicates that only exact matches are wanted, i.e. text in which the given entity and cue are adjacent. A set of cues is manually specified to capture some common and simple ways in which information about the general kind of entity is expressed. For places we used cues like 'is a', 'is famous for', 'is popular with', 'was built'. Although we worked with around 40 cues (including single/plural and present/past forms) it seems that a much smaller number are responsible for returning the majority of high ranking facts; in particular (and perhaps unsurprisingly) the generic 'is' seems most productive. The query may also include a disambiguating term. For example, streets and buildings with the same name may occur in different towns, so we can include a town name in the query outside the double quotes, e.g. <"West Street is popular with" Bridport>. For each query, all the unique snippets returned by the search engine (up to a specified maximum) are processed in the next step; typically a snippet is a few lines of text from a webpage around the words that match the query, often broken in mid-sentence.

II. Shallow Chunk Snippets to Make Candidate Facts. Because all the information that we retrieve about the entity is expressed as 'Entity Cue ...', then we can use a simple extraction pattern to obtain candidate facts from the retrieved snippets. For both English and Latvian the gist of the pattern is 'BOUNDARY ENTITY CUE TEXT-FRAGMENT BOUNDARY', such that 'TEXT-FRAGMENT' captures the 'Text-Fragment' part of a fact. The details of the pattern are captured in a regular expression on a language-specific basis, e.g. to specify boundary words and punctuation, to allow optional words to

appear in between ENTITY and CUE, and to reorder the elements for non-SVO languages. A successful match of the pattern on a snippet leads to the generation of a candidate fact: using the extraction pattern in the Appendix, the snippet text "... in London. Big Ben was named after Sir Benjamin Hall. ..." matches, giving the candidate fact (Big Ben, was named, after Sir Benjamin Hall) but "The square next to Big Ben was named in 1848..." does not match.

III. Filter Candidate Facts. Four filters are used as a quality control, the first two of which require language-specific word lists built manually over a number of runs of the algorithm. *General filter words*: a candidate fact containing any of the given filter words is removed; this can be used to remove potentially subjective statements containing 'me', 'my, 'our', 'amazing', 'fantastic', etc. *Invalid end words*: to catch some erroneous shallow chunking (most likely due to noisy web data, or to a badly cut search engine snippet) this filter removes candidate facts ending in words such 'to', 'from', 'by', etc. *Length of Text-Fragment*: a threshold can be set to filter out candidate facts with text-fragments shorter than the specified number of words; it seems that shorter text-fragments are more likely to lead to incomplete or incorrect facts. *Words all in capitals*: when this filter is turned on, any candidate fact containing a word all in capitals is removed; this is good for removing spam and content in an informal style, but of course it also removes candidate facts containing acronyms.

IV. Score and Sort Facts. Our idea here is to rank facts, at least coarsely, so that we are more likely to get correct and interesting facts at the top. The notions of correctness and interest are each problematic and difficult to unpick for the purposes of algorithm design and evaluation. Here we exploit the overlap between candidate facts for the same Entity-Cue pair to capture these notions to some extent. For each Entity-Cue pair a keyword frequency list is generated by counting the occurrence of all words in the Text-Fragments for that pair; words in a stop word file are ignored. The score for each fact is then calculated by summing the Entity-Cue frequencies of each word in the Text-Fragment, so that facts containing words that were common in other facts with the same Entity-Cue will score highly. If shorter facts are wanted then the sum is divided by the word length of the Text-Fragment. We see two main ways in which the sum score for a fact can get high: (i) there are many overlapping Text-Fragments for an Entity-Cue pair, so there are some high word frequencies; and, (ii) a fact contains more of those high frequency words than other facts. Thus, we hope to get high ranked facts with the most appropriate Cue for the Entity, and the best Text-Fragment for the Entity-Cue pair.

To give an impression of how ranking works, Figure 1 shows the top and bottom 10 facts returned for "Eiffel Tower", using the 'sum only' scoring. The top ranked facts are generally rich in correct information about the given entity. In contrast, incomplete and trivial facts end up low down the list. We see that 4 of the top 10 facts have the Cue 'was built' which seems like a good cue for interesting information about an historical monument. The high-ranking facts with this Cue include words like "Paris", "1889", "international", "exhibition"

```
(Eiffel Tower, was built, in 1889 for an international exhibition in
Paris)
(Eiffel Tower, was named, after an ingenious engineer whose design of the
tower turned it into a reality and pride of the French nation)
(Eiffel Tower, is, an iron tower built during 1887-1889 on the Champ de
Mars beside the Seine River in Paris)
(Eiffel Tower, was one of, the first tall structures in the world to
contain passenger elevators)
(Eiffel Tower, was one of, the landmarks visited by Luigi when he came to
save Paris from invading Koopa Troopas)
(Eiffel Tower, was built, by Gustave Eiffel for the International
Exhibition of Paris of 1889 commemorating the centenary of the French
Revolution)
(Eiffel Tower, was one of, the first structures in the world to have
passenger elevators)
(Eiffel Tower, was built, in 1889 for the Universal Exposition
celebrating the centenary of the French Revolution)
(Eiffel Tower, was built, as a temporary structure for an exhibition in
1889)
(Eiffel Tower, is named, after its designer and engineer Alexandre
Gustave Eiffel)
...
(Eiffel Tower, is built, for the Paris exposition)
(Eiffel Tower, was famous, enough for everyone to know)
(Eiffel Tower, is made, up of a base)
(Eiffel Tower, was made, for the Exposition Universelle)
(Eiffel Tower, is made, of over 10)
(Eiffel Tower, is made, from 18)
(Eiffel Tower, is made, of 3 platforms)
(Eiffel Tower, is made, with 2)
(Eiffel Tower, is famous, throughout the world)
(Eiffel Tower, is famous, for a reason)
```

Fig. 1. The top 10 and bottom 10 facts for the entity "Eiffel Tower"

which are likely to appear after "Eiffel Tower was built..." on many web pages - the fact with all four of these words is ranked highest. For Latvian the top-ranked fact was "Francijas pazīstamākajiem simboliem un gadā to apmeklē aptuveni seši miljoni cilvēku" ("The Eiffel Tower is one of best known symbols of France and it is visited by around 6 million people a year"); this suggests that there is less information about the tower's history available on the Latvian web. For an example of how an undesirable fact is ranked low, see the fact ranked tenth from bottom in Figure 1. This includes the words "Paris" and "exposition" which generally would be highly associated with "Eiffel Tower" but since the fact has the Cue "is built" (rather than "was built") then these words and the fact score low.

To look at how we can select between long and short facts, Figure 2 shows the top ranked facts for a variety of places using the two scoring options described in IV, i.e. 'simple sum', and 'sum divided by number of words'.

```
(North Bridge, was, originally built in 1772 to connect the burgh with
the Port of Leith to the north)
(North Bridge, was built, to link the New Town with the Old Town)
(Bahnhofstrasse, is, where well-heeled bankers and perfectly-coiffed
ladies shop for designer clothing and gold watches)
(Bahnhofstrasse, is, Zurich's main shopping avenue)
(Durdle Door, is a, natural limestone arch on the Jurassic Coast near
Lulworth in Dorset)
(Durdle Door, is a, limestone arch)
(The Matterhorn, was one of, the last Alpine mountains to be ascended due
to its imposing shape and unpredictable weather)
(The Matterhorn, was, first climbed in 1865)
```

Fig. 2. Pairs of facts about places. The first is the top ranked fact with 'simple sum' score, the second is the top ranked with 'sum divided by number of words'.

4.2 Evaluation

This evaluation used 68 place names in English and 60 place names in Latvian from around Europe. We chose an even mixture of urban / rural and famous / not famous places from European cities (London, Riga, Zurich and Dublin) and countryside (UK, Latvia, Switzerland and Ireland), and various types of place - churches, statues, mountains, rivers, etc. For each place the top ranked fact was used for evaluation; see Appendix for the settings used to generate facts.

Evaluating Correctness of Facts. Each of the facts in English was rated as correct or incorrect by an investigator by searching for the fact on the web in the following manner. If the fact was found on Wikipedia, or an official tourist website for the region, and on one other website, or if the fact was found on three independent websites, it was marked as correct. If part of the fact was found on the web using this technique, then the fact was marked as partially correct. Otherwise the fact was marked as incorrect. Due to the lack of coverage in Latvian on the web, Latvian facts were rated as correct if they were located on Wikipedia, or an official tourist website for the region or if they were known to be correct by the investigator. For the English experiment 35 of the 68 facts were marked as correct (51%), 13 were partially correct (19%) and 20 (30%) were incorrect. Analysing the partially correct facts revealed that 11 of the 13 were incomplete facts, e.g.: (Dridzis Lake, is, the deepest lake not only in Latgale); here it looks like our chunking pattern cut too soon (i.e. on the word "but"), although a similar problem occurs occasionally with the way the search engine creates snippets. The other two partially correct facts had spurious material at the end of the fact, e.g.: (Mount Titlis, is the largest, winter sports paradise in Central Switzerland _ even the most demanding skiers); the unusual punctuation '_' is missed by our chunking pattern. Analysing the 20 incorrect facts, we found that only six of them were actually false, for example a fact which was supposed to be about the National Museum in Zurich actually referred to a museum in Prague; this is despite our use of Zurich as a disambiguating term. Eight of the

Table 1. Responses from 5 subjects for 68 English facts, and 60 Latvian facts

	English		Latvian	
	5/5 subjects said 'Yes'	>= 3/5 said 'Yes'	5/5 subjects said 'Yes'	>= 3/5 said 'Yes'
Is this the type of fact you would expect to read in a travel guide?	26/68 (38%)	53/68 (78%)	14/60 (23%)	38/60 (63%)

incorrect facts were unreadable, for example: (Daugava river, is, soon to be a prelude of things to come that would prove 2000 wasn't Cappellini's year) which we put down to web noise. The correctness of the remaining 6 incorrect facts was actually indeterminable, e.g.: (Bastejkalns Park, was, renovated during last winter); we have since added words with temporal reference like 'last' to the filter words list, as well as deictic words like 'this'. Similar to the English results, for the Latvian evaluation 32 of the 60 facts were marked as correct (53%), 19 (32%) were partially correct and 9 (15%) were incorrect.

Evaluating the Interest of Facts. Ten native English speakers were each presented with 34 English facts to rate. Ten native Latvian speakers were each presented with 30 Latvian facts to rate. In this way each fact was rated by 5 subjects. The lists of facts presented to subjects were randomly chosen using a Latin square. For each fact, subjects answered 'yes' or 'no' to the question: "Is this the type of fact you would expect to read in a travel guide?" The question is intended to get at the notion of interest in a way specific to our application scenario, i.e. we assume users would be happy with travel guide like facts added to their image captions. Results are summarised in Table 1 which indicates that, more often than not, our algorithm is producing as its top ranked fact something that most people find acceptable as a fact for something like a travel guide.

Our evaluation criteria for fact correctness were rather strict: note that a majority of subjects rated more facts as interesting (78% English and 63% Latvian) than we ourselves rated as correct (around 50% for both). As noted, it seems that some relatively simple changes to our extraction patterns and word lists will improve our correctness score quite considerably, so overall we are confident that the fundamentals of the approach are sound. Importantly, the approach was very cheap to port between languages. Porting from English to Latvian required only a small modification to the extraction pattern, and the translation of the cue set (see Appendix); other word lists were also translated, but, for Latvian, these had little impact on results.

4.3 Enhancement of Image Captions

Our motivation for doing fact extraction was to add information about places into image captions which provides a scenario for evaluating the utility of the facts that we extract. Sets of 30 image captions were created in English and Latvian for images depicting urban and rural places occurring in Ireland, UK, Latvia

and Switzerland. Half the captions were in the form - 'PLACE photographed in LOCATION.' The other half were in the form - 'Photo taken near PLACE in LOCATION.' Half of the captions also had the time of day inserted into the sentence - 'Photo taken in the TIME_OF_DAY near PLACE in LOCATION'.

Each of the 30 English captions had a fact added in two different ways: (1) insert fact as a subclause in the original sentence; and (2) append the fact to the original caption as a new sentence. This led to 60 enhanced English captions. For (1) the string ', which CUE TEXT-FRAGMENT,' was inserted after the place name in the caption. Recall, we are only able to insert information as a subclause - which keeps the captions more compact - because we have partially structured facts (cf. Section 2). For (2) a second sentence was formed by adding 'PLACE CUE TEXT-FRAGMENT' after the original caption. Insertion as subclause was deemed inappropriate for Latvian, so we had just 30 enhanced Latvian captions with facts added as sentences. We ensured that correct facts were added because we wanted to concentrate on evaluating the readability of the enhanced captions.

The 60 enhanced English captions were presented to 6 native English speakers, in random orders for judgment. The 30 enhanced Latvian captions were presented to 6 native Latvian speakers. For each enhanced caption, subjects answered 'yes' or 'no' to the question: "Does this sentence read naturally to you?". When facts were added as new sentences then a majority of subjects deemed 29/30 (97%) of the enhanced image captions to be readable both for English and for Latvian. The results (English only) for inserting facts as subclauses seemed to depend on the form of the original caption. For 15/15 (100%) of captions with the form "Place photographed in location" a majority of subjects judged the enhanced caption with fact inserted as subclause to be readable. For the other caption form only 7/15 (47%) enhanced captions were judged readable by a majority of subjects. Upon inspection, it seemed that these captions tended to be quite long already (including additional temporal information), so a further subclause, even though grammatically correct, became awkward to read.

5 Conclusions

To summarise, a new kind of fact extraction task was defined, and a solution to the task was evaluated for two rather different languages. It was shown that it is possible to extract useful partially structured facts about different kinds of entity in a broad domain, using a common approach that ports easily between languages in the absence of existing linguistic resources. In contrast with traditional IR techniques we produce output that is more amenable to further automated processing. In contrast with traditional IE techniques our approach has the potential to cover much broader domains and many more languages.

Of course we need to try other kinds of language before making strong claims about portability. Although Latvian is a free word order language, the SVO order does dominate, so we were able to get good results with just one extraction pattern. However, even in languages with more variation in word ordering, we expect that we could use just a few extraction patterns based around cue sets.

What is less clear to us is the ease with which we can port to other domains. Whilst we found interesting facts about many different kinds of places were expressed using a relatively small number of common cues, this may not be the case for all kinds of entities. That said, in preliminary work, we got some encouraging facts about people and organizations using just a few cues.

Beyond the image captioning application and template-based text generation, we see potential for the "Tell Me About..." task in other areas. For some kinds of queries to search engines, users may benefit from being presented with a few facts about their topic of interest: we feel that our chunking of information and ranking of facts can add value to the snippets returned by a search engine. Recently some search websites have started to offer something more like knowledge retrieval on top of information retrieval [13], [14], and our impression is that our kind of fact extraction could contribute to such endeavours.

Acknowledgement. The research reported in this paper is part of the project TRIPOD supported by the European commission under contract No. 045335.

References

1. Purves, R.S., Edwardes, A.J., Sanderson, M.: Describing the Where - improving image annotation and search through geography. In: First Intl. Workshop on Metadata Mining for Image Understanding (2008)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, New York (1999)
3. Salton, G., Allan, J., Buckley, C.: Approaches to passage retrieval in full text information systems. In: 16th ACM SIGIR, pp. 49–58 (1993)
4. Sarawagi, S.: Information Extraction. Foundations and Trends in Databases 1(3), 261–377 (2008)
5. Lin, J.: An Exploration of the Principles Underlying Redundancy-based Factoid Question Answering. ACM Trans. Information Systems 25(2), 1–55 (2007)
6. Dumais, S., et al.: Web Question Answering: Is More Always Better? In: 25th ACM SIGIR, pp. 291–298 (2002)
7. Goldstein, J., et al.: Multi-document Summarization by Sentence Extraction. In: NAACL-ANLP 2000 Workshop on Automatic Summarization, pp. 40–48 (2000)
8. Pasca, M., et al.: Organizing and Searching the World Wide Web of Facts - Step One: the One-Million Fact Extraction Challenge. In: 21st Nat. Conf. on AI (AAAI 2006), pp. 1400–1405 (2006)
9. Banko, M., Etzioni, O.: The Tradeoffs Between Open and Traditional Relation Extraction. In: ACL 2008, pp. 28–36 (2008)
10. Etzioni, O., et al.: Open Information Extraction from the Web. Comms. of the ACM 51(12), 68–74 (2008)
11. TextRunner Search (March 30, 2010), http://www.cs.washington.edu/research/
12. Yahoo! Search BOSS (March 30, 2010), http://developer.yahoo.com/search/boss/
13. Powerset (March 30, 2010), http://powerset.com
14. Google Squared (March 30, 2010), http://www.google.com/squared

Appendix: Settings Used for Evaluation Runs

English

Cues used in queries to search engine: `is, was, is the, was the, is a, was a, is an, was an, is in, is on, is by, is next to, is near to, is known, is famous, is located, is one of, was built, is made of, is named, was named, is home to, was home to, is used, was used, was completed, was destroyed, was damaged, is the site of, was the site of, was the scene of, was made famous, is the most, is the biggest, is the largest, is the smallest, is the oldest, can be seen from, is popular, is popular with, features, offers, is located by, is located on, is located in, is famous for, is known for, was built by, was built in, was built for, was built to, is open`

Regular Expression for Shallow Chunking of Snippets: `^|\.|\,|\;|\:|\?|\!|(the|The)\s*ENTITY\s*CUE\s*(.*?)(\.|\,|\;|\:|\?|\!)|((\b(and)\b|\b(but)\b)))`
ENTITY and CUE are interpolated at run-time, (.*?) captures the Text-Fragment.
Filter words: `I, my, me, mine, you, your, yours, we, us, ours, another, recently, this, also, other, further, must, should, could, sensational, fun, deserves, excellent, amazing, wonderful, miles, kilometres, m, km, minutes, min, mins, hours, hour, probably, actually, possibly`
Scoring stop words: `the, of, is, for, a, an, and`
Invalid final words: `a, the, those, these, with, by, and, but, which, that, for, like, as`

Latvian

Cues used in queries to search engine: `ir, bija, ir pazīstams, ir pazīstama, ir slavens, ir slavena, ir ievērojams, ir ievērojama, atrodas, ir viens no, ir viena no, ir blakus, ir netālu no, tika uzcelts, tika uzcelta, tika celta, tika celts, ir veidots no, ir veidota no, ir izgatavots no, ir izgatavota no, ir nosaukts, ir nosaukta, tika nosaukts, tika nosaukta, bija mājas, tika lietots, tika lietota, tika pabeigts, tika pabeigta, tika sagrauta, tika sagrauts, ir pats, ir pati, ir lielākais, ir lielākā, ir mazākais, ir mazākā, ir vecākais, ir vecākā, ir garākā, ir dzilākā, ir dzilākais, ir augstākais, var redzēt no, ir redzams no, ir redzama no, atklāj, ir raksturīgs ar, ir raksturīga ar`

Regular Expression for Shallow Chunking of Snippets: `ENTITY\s*CUE\s*(.*?)(\.|\!|\;|\:|\?|\[)`

For both languages the maximum snippets returned from search engine for a single query was 20, scoring metric was 'simple sum', and score threshold = 3.

Passage Retrieval in Log Files: An Approach Based on Query Enrichment

Hassan Saneifar[1,2], Stéphane Bonniol[2], Anne Laurent[1], Pascal Poncelet[1], and Mathieu Roche[1]

[1] LIRMM - Univ. Montpellier 2 - CNRS, France
[2] Satin Technologies, France
{saneifar,laurent,poncelet,mroche}@lirmm.fr,
{stephane.bonniol}@satin-tech.com
http://www.lirmm.fr/~{saneifar,laurent,poncelet,mroche}

Abstract. The question answering systems are considered the next generation of search engines. This paper focuses on the first step of this process, which is to search for relevant passages containing answers. Passage Retrieval, can be difficult because of the complexity of data, log files in our case. Our contribution is based on the enrichment of queries by using a *learning method* and a *novel term weighting* function. This original term weighting function, used within the enrichment process, aims to assign a weight to terms according to their relatedness to the context of answers. Experiments conducted on *real data* show that our protocol of primitive query enrichment make it possible to retrieve relevant passages.

Keywords: Information Retrieval, Question Answering System, Passage Retrieval, Query Enrichment, Context Learning.

1 Introduction

Information Retrieval (IR) aims to find documents related to a topic specified by a user. The topic is normally expressed as a list of specific terms. However, the needs of some application domains make Information Retrieval (IR) methods inefficient. Indeed, when the goal is to find specific and concise answers, Information Retrieval systems are not relevant according to the considerable amount of documents that they retrieve as possibilities. Moreover, the information found in retrieved documents is not always correlated with the given query. That is why Question Answering (QA) systems are an important research topic nowadays. Question Answering systems aim to find a relevant fragment of a document which could be regarded as the best possible concise answer to a question given by user. There are two main categories of QA systems: (1) Open domain and (2) Restricted domain.

In the first case, questions arise about general domains and sources of information are large corpora consisting of documents of several fields (e.g. corpus of web pages). The evaluation of open domain QA systems has been built since 1999 in TREC[1] (Text REtrieval Conference) American evaluation campaigns.

[1] http ://trec.nist.gov/

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 357–368, 2010.

In the second category, QA systems are designed to answer questions in a specific area. In this kind of QA systems, information resources are technical and specialized documents. The restricted domains, called also closed domains, have certain characteristics which make the methods of open domain QA become less useful [3]. We detail these characteristics and some features of specialized textual data which make answer retrieval more difficult in Sect. 2.

The Passage Retrieval represents an important phase of QA process. To give an efficient and reliable definition, a passage is defined as a fixed-length sequence of words which can begins and ends anywhere in a document [9]. The Passage Retrieval is the task of searching for passages which may contain the answer to a given question.

In this paper, we present our work on passage retrieval in a specialized domain. We deal with a particular type of complex textual data which are log files generated by Integrated Circuit (IC) design tools. These log files are digital reports on configurations, conditions, and states of systems. In this area, checking the quality of products requires to answer some technical and specialized questions. At this stage, our goal is to find segments of the logs that contain the answers to the questions of quality check. The particularity of such textual data (*i.e. log files*) and characteristics of restricted (closed) domains impact significantly the accuracy and performance of passage retrieval in this context.

We propose in this paper, a passage retrieval system based on a new approach of query enrichment. Our query enrichment process is based on a *learning approach* and a *new weighting function* which gives a score to terms of corpus according to their *relatedness* to the *context of answers*. The enrichment process takes place in two phases. First, we propose a method for learning the context of questions, based on the notion of "lexical world". In the learning phase we identify the terms representing the context of questions. Then, the initial queries are enriched by these terms. Secondly, we propose an *original term weighting function* which aims at giving a high score to terms of corpus which have a significant probability to exist in the relevant passages. The terms having the highest scores are included in the query in order to improve its enrichment. Our approach is an interactive system based on relevant feedback. We show that our approach gives *satisfactory* results on *real data*.

In Section 2, we present the specific characteristics of log files and also the limits of QA systems in restricted domains. Existing work concerning the QA systems are presented in Sect. 3. Section 4 presents some notions used in enrichment processes and also the first phase of query enrichment. In Section 5 we develop our approach of passage retrieval and query enrichment by presenting our novel term weighting function. Experiments on real data are presented in Sect. 6.

2 Problem Study

Log files generated by Integrated Circuits design tools are not systematically exploited in an efficient way despite the fact that they contain the essential

information to evaluate the design quality. The particular characteristics of logs, described below, make classical techniques of Natural Language Processing (NLP) and Information Retrieval irrelevant.

2.1 Information Retrieval and Log Files

We consider log files as a kind of "complex textual data", i.e. containing *multi-source*, *heterogeneous*, and *multi-format* data. Indeed, in the design of Integrated Circuits, different design tools can be used in the same time while each tool generates its own log files. Therefore, despite the fact that the logs of the same design level contain the *same* information, their *structures* and *vocabulary* can vary significantly *depending* on the *used design tool*. More precisely, in order to report the same information, each design tool uses its own vocabulary.

In this domain the questions (queries) are expressed using a vocabulary that does not necessarily correspond to the vocabulary of all tools. However, a system should be able to answer the questions regardless of the type of tools that generated the log files. We explain this issue with an example. Consider the sentence "`Capture the total fixed STD cell`" as a given question (query). We produce two log files (eg., log "A" and log "B") by two different tools. The answer to the question, in log "A", is expressed as follows:

```
Std cell area: 77346 sites
non-fixed:74974 fixed:2372
```

While the answer, in log "B", is expressed in this line:

```
preplaced standard cell is: 24678
```

As shown above, the same information in two log files produced by two different tools, is represented by different structures and vocabulary. The keywords of the question (i.e. `Fixed, STD` & `cell`) exist in the answer extracted from log "A" while the answer from log "B" contains only the word "`cell`". Insofar as there is a dictionary associating the word "`STD`" with "`standard`", we can also consider the word "`standard`". However, by giving these two words as a query to an information retrieval system, irrelevant passages of log "B" are retrieved:

```
standard cell seeds is : 4567
Total standard cell length = 0.4536
```

This can be explained by the fact that the question is expressed using the vocabulary of log "A" which is different from the vocabulary of log "B". In other words, for a given question, the *relevant answers* found in the logs of *some tools* do *not* necessarily contain the *keywords* of the question. Therefore, the initial query (*created by taking the keywords of question*) may be relevant to logs generated by a tool, but irrelevant to logs generated by another tool[2] whereas we aim to answer questions regardless type of tools generating log files.

The *existence of question keywords* (or their syntactic variants) in a *passage* is an important factor to assess the *relevance* of the passage. The approaches

[2] While all of these logs report the same information using different vocabularies.

which are based on the notion of common terms between questions and passages are detailed in Sect. 3.

Moreover, the performance of a QA system depends largely on redundant occurrences of answers in the corpus in which answers are seek [1] [7]. The methods developed for QA systems are generally based on the assumption that there are several instances of answers in corpus. But information is rarely repeated in the log files of IC design tools. This means that for a question, there is only one occurrence of the answer in the corpus and thus one relevant passage (containing the answer).

In addition, design tools change over time, often unexpectedly. Therefore, the *format of the data* in the log files changes, which makes automatic data management difficult. Moreover, the language used in these logs is a difficulty that impacts information extraction methods. Although the language used in these logs is English, their contents do not usually comply with "*classic*" grammar. In the processing of log files, we also deal with multi-format data: textual data, numerical data, alphanumerical, and structured data (e.g., table and data block). There are also many technical words that contain special characters which are only understandable considering the domain documentation. Due to these specific characteristics of log files, NLP and IR methods, developed for texts written in natural language, are not necessarily well adapted to log files.

We therefore suggest the enrichment of initial queries in order to make them relevant to all types of logs (*generated by any kind of design tools*). We explain the query enrichment process in Sect. 4.

2.2 Passage Retrieval in Log Files

The passages retrieval phase influences significantly the performance of QA systems because final answers are sought in the retrieved passages. Most QA systems for a given question, extract a large number of passages which likely contain the answer. But an important point in QA systems is to limit, as much as possible, the number of passages in which the final answer extraction is performed. Since we are situated in a very specialized domain, high precision in the final answers (i.e. the percentage of correct answers) is a very important issue. This implies that the passage retrieval system has to classify relevant passages (based on a relevance score) in the top positions among all retrieved candidate passages.

3 Related Work

Most passage retrieval algorithms depend on occurrences of query keywords in corpus [9]. To enhance the query, the use of morphological and semantic variants of query keywords is studied in [2].

The reformulation of questions (also referred to as *surface patterns* and *paraphrases*) is a standard method used to improve the performance of QA. The technique is based on identifying various ways of expressing an answer given a natural language question [5]. For example, for a question like "*Who founded the*

American Red Cross?", QA systems based on surface patterns seek reformulations like "*the founder of the American Red Cross is X*" or "*X, the founder of the American Red Cross*". The question reformulation using surface patterns is also exploited in TREC9 and TREC10. [8] and [5] present different approaches to take semantic variations (semantic reformulations) into account in order to complement the syntactic variants. To find relevant passages, [6] evaluates each passage using a scoring function based on the coverage of "question keywords" which exist also in the passage. QA systems also use query expansion methods to improve performance of retrieval. These methods can use the thesaurus [4] or be based on the incorporation of the most frequent terms in the m relevant documents.

Despite the satisfactory results achieved by the use of surface patterns and syntactic variants in mentioned work, these methods are irrelevant in the context of log files according to the problems described in Section 2. Indeed, the main reasons for irrelevancy of such methods are related to the fact that an answer is not reformulated in different ways in a corpus of log files. Also, there is a lack of redundancy of answers in corpus of logs. In addition, there are several technical and alphanumeric keywords in the domain of log files for which the use of syntactic or semantic variants appears to be complex or unmeaning.

4 Passage Retrieval Based on Query Enrichment

We propose in this paper, a passage retrieval approach based on a new interactive process of query enrichment. The enrichment of query is based on a context learning process and is associated with a novel and original term weighting function. Our protocol of context learning is designed to determine the context of a given question by analyzing the terms [3] co-occurring around the question keywords in the corpus. The new term scoring function proposed in this paper, identifies the terms which are related to the answers.
The architecture of our approach consists of three main modules:

1. Enrichment of the initial query by context learning
2. Enrichment by terms which are likely related to answer
3. Passage retrieval using the enriched query

The first module enriches the initial query extracted from a question in natural language. This module aims at making the initial query relevant to all types of logs which are generated by different tools. At this step, by learning the context of question, we enrich the initial query by the most significant terms of the context.

The second module is activated for a second enrichment of the query in order to obtain a higher accuracy. At this phase, we aim at identifying the terms which are likely related to answer in order to integrate them in the query. For

[3] In this paper, the word "term" refers to both words and multi-word terms of the domain.

this purpose, we propose a process of scoring of terms based on a new weighting function. The weighting function is designed to give a score to terms according to their relatedness to answers. We devote Section 5 to this topic.

In the third module, we seek the relevant passages in the logs generated by a tool different from the one which has been used in the learning phase. That is, we have two different corpora of logs. The first one (called *training corpus*) is used in learning phase and the second corpus (called *test corpus*) is the corpus in which we retrieve the passages relevant to given questions. The logs of the test corpus have structures and a vocabulary significantly different from the logs of the training corpus. In our approach, we look for specialized context (called hereafter "*lexical world*") of question keywords. In order to characterize and present in a relevant way the specialized context of keywords, we use the terminological knowledge extracted from logs. Before explaining the query enrichment processes, we develop the concept of "lexical world" and the use of terminological knowledge in the following subsections.

4.1 Lexical World

The lexical world of a term is a small fragment of document in which the term is seen. We consider this fragment specialized context of the term because terms located around a term (within a small fragment) generally have a strong semantic and/or contextual relations. Therefore, by determining the *lexical world* of a term in a document, we identify *terms* that *tend* to *appear around it* in that document. We do not put any limit on the size of lexical worlds (eg., a few lines, a few words, etc.) as it have to be determined pragmatically based on the type of documents.

In order to present the lexical world of a term, several solutions are possible. As a first solution, we characterize the lexical world of a term by a set of "words" (*called also bag of words*) which are located around the term and present "Noun", "Verb", or "Adjective" parts of speech. As a second solution, the lexical world of a term is presented by co-occurring words like bigrams of words (i.e. any two adjacent words) or multi-word terms (few adjacent words forming a meaningful term) which are seen around the term. We detail this point in the next section.

4.2 Terminological Knowledge

As mentioned above, the lexical worlds can be characterized in different ways: By "words", "multi-word terms", or "bigrams of words". According to our experiments, the multi-word terms and words are more representative than bigrams of words. Hence, we create two types of lexical world: (1) Consisting of words and (2) Consisting of multi-word terms and words. In order to determine the multi-word terms, we extract the terminology of logs using the method presented in [13]. This method adapted to the specific characteristics of logs, extracts the multi-word terms according to syntactic patterns in the log files. To choose the relevant and domain-specific terms, we use the terminology validation and filtering protocol presented in [12]. We have finally the valid and relevant multi-word terms to characterize lexical worlds.

4.3 Query Enrichment by Context Learning

We explain here the first module of our query enrichment approach. For a given question, we look initially to learn the context of the question and characterize it by its most significant terms. These terms represent at best the context of the question. Thus, it is expected that the passages corresponding to the question share some of these terms regardless of different kind of log files.

Firstly, we seek lexical worlds of question keywords. The found lexical worlds and the initial query are vectorized. We use an IR system based on Vector Space (VS) model in order to select the lexical world most correlated to the initial query. Then, we choose the most representative terms of selected lexical world. For this purpose, we select the n terms having the highest tf-idf scores [11]. We get the first enriched query by inserting the selected terms in the initial one.

Since the next phase of query enrichment based on the novel weighting function is the main contribution of this paper, we develop it in Section 5. Thus, we explain in the following subsection how we look for relevant passages once initial queries are enriched.

4.4 Passage Retrieval in Log Files

We detail here the process of finding relevant passages on the test corpus of log files. First, we segment the logs of the test corpus. Segmentation is performed according to the structure of the text like data blocks, tables, separating lines, etc. Each segment is seen as a passage of log files containing potentially the answer. Second, we enrich the initial query in order to make it relevant to all types of log files. We remind that we aim at adapting the initial query to vocabulary of all types of log files by our query enrichment approach. Third, we build a system of IR in order to find the relevant passages. The IR system uses an indexing function and a similarity measure. In this phase, we experiment our IR system by using *tf-idf* and Binary representation as indexing functions. The Cosine and Jaccard measures are used as a similarity measure. The IR system gives a relevancy score to every passage (segment) according to the enriched queries. Then, we order the passages based on their relevancy score and propose the top-ranked passages to the user.

Several approaches of passage retrieval return a considerable number of candidate passages. Our experiments conducted on real data assert that in more than 80% of the cases, the relevant passage is located among the three top-ranked passages. That is, our approach often ranks the relevant passage among the three top candidate passages returned by a system as possible results.

5 How to Find Terms Correlated to Answers

To improve the relevance of the query to different types of logs, we propose a second module of enrichment of the query. In this module we have as input the query enriched in the phase of context learning and the test corpus of logs in

which we seek the relevant passages (different from the training corpus which is used in the context learning phase).

Our motivation is to find the terms in the logs of the *test corpus*, which are likely to exist in the relevant passage and are, therefore, related to the answer. For this purpose, we offer here a term selection process based on a new term weighting (scoring) function. Terms will be selected based on their score obtained by this scoring function. This function gives a score to each term based on three assumptions:

- the most *correlated terms* to the answer are in a *lexical world* representing the *context* of the question
- the final query must contain *discriminant terms* (i.e. terms which do not exist in several passages)
- *most of the terms of a relevant query* are associated with the corresponding *relevant passage* (i.e. the relevant passage contains most of query keywords)

Based on the first assumption, for each query keyword, we extract their lexical worlds in the *test corpus*. We note that the system has no information on the logs of the test corpus and relevant passages[4]. We finally obtain a set of lexical worlds, corresponding to the query keywords, which are extracted from the test corpus. Our scoring function is based on the two last assumptions.

Firstly, we seek to select discriminative terms (i.e. terms that do not exist in several lexical worlds). In other words, we look for terms with a very low frequency of occurrence in different lexical worlds. For this, we use the *idf* (inverse document frequency) function by considering each lexical world extracted in the previous step as a document and all lexical worlds as the corpus. In this way, we favour terms which exist in one or a very small number of lexical worlds.

Secondly, in order to favour the terms which likely exist in the relevant passage, we give another score (*besides idf*) to each term of lexical worlds based on the third assumption. For a given term T, this score that we call LWF (Lexical World Frequency), depends on the number of query keywords which are associated with the lexical world to which the term T belongs. This score presents a form of *df* (document frequency) where a document corresponds to all of lexical worlds associated with a query keyword. Indeed, by this score, we measure how important is the lexical world in which the given term is located. This importance is calculated according to the number of query keywords which are associated with the lexical world. The LWF formula for the term i existing in the lexical world j is calculated as follows:

$$\mathbf{lwf_{ij}} = \mathbf{1}/\log(\mathbf{M}/\mathbf{n_j})$$

M is the total number of query keywords and n_j shows the number of query keywords which are associated with the lexical world j.

[4] The learning process is performed on the logs of the training corpus which are generated by a tool using a significantly different vocabulary and structures from the tool generating the logs of the test corpus.

The final score of a term that we call TRQ (Term Relatedness to Query) is calculated using the following formula:

$$\mathbf{TRQ} = \alpha * (\mathbf{lwf}) + (\mathbf{1} - \alpha) * \mathbf{idf}$$

According to the experiments, the most relevant value of α is 0.25[5]. This means that we give more weight to the frequency of terms in the corpus and a smaller weight but which influences the final results to lwf.

We explain with an example, the process of selection of terms which are likely related to the answer. Supposing Q=$\{W_a, W_b, W_d\}$ as a query enriched by the first module (learning phase) and $\log_b$ as a log file containing seven segments:

$\mathbf{S_1}$=$\{W_a\ W_k\ W_m\ W_b\}$ $\mathbf{S_4}$=$\{W_a\ W_c\ W_e\ W_q\}$ $\mathbf{S_7}$=$\{W_b\ W_c\ W_k\}$
$\mathbf{S_2}$=$\{W_d\ W_k\}$ $\mathbf{S_5}$=$\{W_b\ W_e\}$
$\mathbf{S_3}$=$\{W_z\}$ $\mathbf{S_6}$=$\{W_z\}$

We consider that the border of lexical worlds (*border of the selected fragment of text around a given term*) corresponds to the border of segments (i.e. a lexical world is not bigger than the corresponding segment). Among sept segments, there are fives which are associated with terms of Q. Thus, we obtain S_1, S_2, S_4, S_5, S_7 as the set of lexical worlds of question keywords. The following lists shows the lexical worlds associated to each keywords of the question[6].

W_a:$\{S_1, S_4\}$ W_b:$\{S_1, S_5, S_7\}$ W_d:$\{S_2\}$

Here, for instance, the word W_a in the query Q is associated with two lexical worlds (S_1 and S_4). The idf score of the word W_k, for example, is equal to $log(5/3) = 0.22$ because, the word W_k exists in three lexical worlds (S_1, S_2 and S_7) among fives. The value of lwf for the word W_k in the segment S_1 is calculated as following: $lwf_{2,1} = 1/\log(3/2) = 5.8$. Indeed, there are two words in the query Q (i.e. W_a and W_b) which are associated with the lexical world S_1 (the lexical world in which the word W_k is located). We note that for a given term, the value of lwf depends on the lexical world in which the given term is located. For example, the value of lwf for the word W_k located in segment S_2 is equal to $lwf_{2,2} = 1/\log(3/1) = 2.1$ as there is just one keyword of the query Q associated to S_2. This means that W_k located in the segment S_2 is less significant (less related to the query) than when it is located in the segment S_1.

Once the TRQ score of all terms of lexical worlds are calculated, we identify the k *highest scores* and select the terms having these scores. However, among the terms selected based on their TRQ scores, there are terms having the same score. To distinguish these terms, we assess their *tendency* to appear *close* to the keywords of the initial query. In other words, we seek to calculate the *dependence* of *selected terms* to the *keywords* of question in the context. For this purpose, we choose the "Dice" measure. This statistical measure has a good performance

[5] We justify the selected value of α by presenting some results in http://www.lirmm.fr/~saneifar/experiments/TRQ.pdf

[6] Since a word can be used in different contexts (i.e. different fragments of document), it can be associated with several lexical worlds.

for tasks of text mining [10]. For a term T and a query keyword W, we calculate the Dice measure as following:

$$Dice(T,W) = \frac{2 * \mid (T,W) \mid}{\mid T \mid + \mid W \mid}$$

For us, $\mid (T,W) \mid$, number of times T and W occur together, corresponds to the number of times where T and W are located in the same line in the corpus of logs. $\mid x \mid$ shows the total number of occurrences of x in the corpus. Finally, the value of Dice measure allows to distinguish the terms obtained in the previous step, which have equal TRQ score. The final score of these terms is obtained by the sum of Dice value and the TRQ score. Note that we select at first the terms according to their TRQ score (i.e. terms having highest TRQ score are selected). If we have the terms having the same TRQ score, we distinguish them by calculating their Dice values.

Finally, as described above, the system ranks the terms based on their TRQ scores (considering Dice values for terms having the same score). The system recommends to the user the k top-ranked terms in order to enrich the query. Our system is also able to integrate automatically the k top-ranked terms into the query in an autonomous mode. The enriched query EQ_{mod2} will be used in passage retrieval.

6 Experiments

We test the performance of our approach on a corpus of log files from the real industrial world (data from the Satin IP company). There are 26 questions which are expressed in natural language and are extracted from standard check-list. Log files are segmented according to their structures (blank lines, tables, data blocks, etc.). Each segment is potentially a relevant passage. Note that for a given question, there is only one relevant passage (*segment*) in the corpus (i.e. there is just one occurrence of the answer in the corpus). The relevance of passages is evaluated as whether the final answer is situated in the passage.

The test corpus that we use for the experiments contains 625 segments and is about 950 KB. Each segment consists of approximately 150 words. The training corpus used in the phase of learning the context of questions consists of logs reporting the same information as logs of the test corpus, but generated by a totally different tool – *thus different vocabulary and segmentations*. For a given question, initial query is obtained by taking the question keywords.

In each category, we experiment our approach while the IR system uses different indexing functions and similarity measures. In order to evaluate the performance of our approach in these different conditions, we use *Mean Reciprocal answer Rank* (MRR) used in TREC as the measure of performance evaluation [14]. This measure takes into account the rank of the correct answer (among the ranked list of candidate answers).

$$MRR = \frac{1}{nb(Question)} \sum_{i=1}^{nb(Question)} \frac{1}{rank(answer)}$$

Table 1. Percentage of question %**P**(**n**) for which the relevant passage is ranked as "**n**". **(a)** performance obtained by using the ***not enriched queries*** (initial queries), **(b)** performance obtained by using the ***enriched queries***.

(a)

	tf-idf		Binary	
	Cos	Jac	Cos	Jac
$\%P(1)$	9	8	11	8
$\%P(2)$	5	4	3	3
$\%P(3)$	4	3	2	2
MRR	0.51	0.50	**0.58**	0.48

(b)

	tf-idf		Binary	
	Cos	Jac	Cos	Jac
$\%P(1)$	20	18	19	13
$\%P(2)$	3	1	0	4
$\%P(3)$	1	3	2	1
MRR	**0.83**	0.78	0.79	0.65

In the experiments, we measure the performance of passage retrieval using the enriched queries. We aim at comparing the performance of passage retrieval using the enriched queries with the performance of passage retrieval using the initial queries (not enriched). That shows how our query enrichment approach improves the relevance of initial queries. Tables 1a, 1b present respectively the results of the performance of passage retrieval using the not enriched query and the enriched ones. $\%P(n)$ is the *percentage of questions* for which the *relevant passage* is *ranked as* n among the retrieved candidate passages as possibilities. Here, we show the results for the three first ranks. As shown in Tab. 1a, by using the not enriched queries, we obtain a MRR value equal to 0.58 in best conditions. While by enriching the initial queries as mentioned in this paper, the MRR improves significantly and reaches 0.83 in best conditions. According to the results, in the best configuration of the IR module and by *using the enrichment of queries*, the relevant passage is ranked in 76% of cases as the first passage among the candidate passages returned by the system. Also, in 92% of cases, the relevant passage is located (ranked) among the three top-ranked passages returned by the system when there are about 650 passages in the corpus.

7 Conclusions

We have presented a process of double enrichment of initial queries in order to improve the performance of passage retrieval in log files. The heterogeneity of the vocabulary and structures of log files and the fact that the keywords used in the questions expressed in natural language do not exist necessarily in the logs make the passage retrieval difficult. Despite these characteristics, our approach makes it possible to adapt an initial query (i.e. list of question keywords) to all types of corresponding log files whatever is their vocabulary. According to the results, by our query enrichment protocol which is based on our novel weighting function called TRQ (Term Relatedness to Query), we obtained a value of MRR equal to 0.83 while the value of MRR was equal to 0.58 by using the *not* enriched queries. We plan to evaluate our system with other models of Information Retrieval. Improving the new term weighting function used in the second phase of query enrichment, represents a major point in the future work.

References

1. Brill, E., Lin, J., Banko, M., Dumais, S., Ng, A.: Data-intensive question answering. In: Proceedings of the Tenth Text REtrieval Conference (TREC), pp. 393–400 (2001)
2. Chalendar, G., Dalmas, T., Elkateb-Gara, F., Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G., Monceaux, L., Robba, I., Vilnat, A.: The question answering system qalc at limsi, experiments in using web and wordnet. In: TREC (2002)
3. Doan-Nguyen, H., Kosseim, L.: The problem of precision on restricted-domain question answering. In: Proceedings the ACL 2004 Workshop on Question Answering in Restricted Domains (ACL-2004), Barcelona, Spain (July 2004)
4. Jing, Y., Croft, W.B.: An association thesaurus for information retrieval. In: RIAO 1994: Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications), pp. 146–160 (1994)
5. Kosseim, L., Yousefi, J.: Improving the performance of question answering with semantically equivalent answer patterns. Data Knowl. Eng. 66(1), 53–67 (2008)
6. Lamjiri, A.K., Dubuc, J., Kosseim, L., Bergler, S.: Indexing low frequency information for answering complex questions. In: RIAO 2007: 8th International Conference on Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications). Carnegie Mellon University, Pittsburgh (2007)
7. Lin, J.: An exploration of the principles underlying redundancy-based factoid question answering. ACM Trans. Inf. Syst. 25(2), 6 (2007)
8. Mollá, D.: Learning of graph-based question answering rules. In: Proc. HLT/NAACL 2006 Workshop on Graph Algorithms for Natural Language Processing, pp. 37–44 (2006)
9. Ofoghi, B., Yearwood, J., Ghosh, R.: A semantic approach to boost passage retrieval effectiveness for question answering. In: ACSC 2006: Proceedings of the 29th Australasian Computer Science Conference, pp. 95–101. Australian Computer Society, Inc., Darlinghurst (2006)
10. Roche, M., Kodratoff, Y.: Text and Web Mining Approaches in Order to Build Specialized Ontologies. Journal of Digital Information 10(4), 6 (2009)
11. Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval. Tech. rep., Ithaca, NY, USA (1987)
12. Saneifar, H., Bonniol, S., Laurent, A., Poncelet, P., Roche, M.: Mining for relevant terms from log files. In: KDIR 2009: Proceedings of International Conference on Knowledge Discovery and Information Retrieval, Madeira, Portugal (October 2009)
13. Saneifar, H., Bonniol, S., Laurent, A., Poncelet, P., Roche, M.: Terminology extraction from log files. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 769–776. Springer, Heidelberg (2009)
14. Voorhees, E.M.: The trec-8 question answering track report. In: Proceedings of TREC-8, pp. 77–82 (1999)

Part-of-Speech Tagging Using Parallel Weighted Finite-State Transducers

Miikka Silfverberg and Krister Lindén

Department of Modern Languages
University of Helsinki
Helsinki, Finland
{miikka.silfverberg,krister.linden}@helsinki.fi

Abstract. We use parallel weighted finite-state transducers to implement a part-of-speech tagger, which obtains state-of-the-art accuracy when used to tag the Europarl corpora for Finnish, Swedish and English. Our system consists of a weighted lexicon and a guesser combined with a bigram model factored into two weighted transducers. We use both lemmas and tag sequences in the bigram model, which guarantees reliable bigram estimates.

Keywords: Weighted Finite-State Transducer, Part-of-Speech Tagging, Markov Model, Europarl.

1 Introduction

Part-of-Speech (POS) taggers play a crucial role in many language applications such as parsers, speech synthesizers, information retrieval systems and translation systems. Systems, which need to process a lot of data, benefit from fast taggers. Generally it is easier to find faster implementations for simple models than for complex ones, so simple models should be preferred, when tagging speed is crucial.

We demonstrate that a straightforward first order Markov model, is sufficient to obtain state-of-the-art accuracy when tagging English, Finnish and Swedish Europarl corpora [Koehn 2005]. The corpora were tagged using the Connexor fdg parsers [Järvinen et al. 2004] and we used the tagged corpora both for training and as a gold standard in testing. Our results indicate that bigram probabilities yield accurate tagging, if lemmas are included in POS analyzes.

Our model consists of a weighted lexicon, a guessing mechanism for unknown words, and two bigram models. We analyze each word in a sentence separately using the weighted lexicon and guesser. The analyzes are then combined into one acyclic minimal weighted finite-state transducer (WFST), whose paths correspond to possible POS analyzes of the sentence. The paths in the sentence WFST are re-scored using the bigram models.

The bigram models assign weights for pairs of successive word forms and corresponding POS analyzes including lemmas. One of the models assigns weight for POS analyzes of word form bigrams starting at even positions in the sentence

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 369–380, 2010.

and the other one assigns weights for bigrams starting at odd positions. Both bigram models are implemented as WFSTs.

The sentence WFST and bigram model WFSTs are combined using weighted intersecting composition [Silfverberg and Lindén 2009], which composes the sentence WFST with the simulated intersection of the bigram models. Finally the POS analysis of the sentence is obtained using a best paths algorithm [Mohri and Riley 2002]. The WFSTs and algorithms for parsing were implemented using an open source transducer library HFST [Linden et al. 2009].

The paper is structured as follows. We first review earlier relevant research in section 2. We then formalize the POS tagging task in section 3 and present our model for a POS tagger as an instance of the general formulation in section 4. In section 5 we demonstrate how to implement the model using WFSTs.

The remainder of the paper deals with training and testing the POS tagger. We present the corpora and parsers used in training and tests in section 6, describe training of the model in section 7, evaluate the implementation in section 8 and analyze the results of the evaluation and present future research directions in section 9. Lastly we conclude the paper in section 10.

2 Previous Research

Statistical POS tagging is a common task in natural language applications. POS taggers can be implemented using a variety of statistical models including Hidden Markov Models (HMM) [Church 1999] [Brants 2000] and Conditional Random Fields [Lafferty et al. 2001].

Markov models are probably the most widely used technique for POS tagging. Some older systems such as [Cutting 1992] used first order models, but the accuracies reported were not very good. E.g. [Cutting 1992] report an accuracy of 96 % for tagging English text. Newer systems like [Brants 2000] have used second order models, which generally lead to better tagging accuracy. [Brants 2000] reports accuracy of 96.46% for tagging the Penn Tree Bank. More recent second order models further improve on accuracy. [Collins 2002] reports 97.11% accuracy and [Shen et al. 2007] 97.33% accuracy on the Penn Tree Bank.

We use lemmas in our bigram model as did [Thede and Harper 1999], who used lexical probabilities in their second order HMM for tagging English and obtained improved accuracy (96% – 97%) w.r.t. a second order model using plain tag sequences. In contrast to this, our model uses only bigram probabilities and it is not an HMM, since we only use frequency counts of POS analyzes for word pairs. In addition we split our bigram model into two components, which reduces its size thus allowing us to use a larger training material.

The idea of syntactic parsing and POS tagging using parallel finite-state constraints was outlined by [Koskenniemi 1990]. The general idea in our system is the same, but instead of a rule-based morphological disambiguator, we implement a statistical tagger using WFSTs. Still, hand-crafted tagging constraints could be added to the system.

3 Formulation of the POS Tagging Task

In this section we formulate the task of Part-of-Speech (POS) tagging and describe probabilistic POS taggers formally.

By a sentence, we mean a sequence of syntactic tokens $s = (s_1 \ ... \ s_n)$ and by a POS analysis of the sentence s, we mean a sequence of POS analyzes $t = (t_1, \ ..., \ t_n)$. We include lemmas in POS analyzes. For each i, the analysis t_i corresponds to the token s_i in sentence s. We denote the set of all sentences by S and the set of all analyzes by T.

A *POS tagger* is a machine which associates each sentence s with its most likely *POS analysis* t_s. To find the most likely POS analyzes for the sentence s, the model estimates the probabilities for all possible analyzes of s using a distribution P. For the sentence s and every possible POS analysis t, the distribution P associates a probability $P(t, s)$. Keeping t fixed, the mapping $s \mapsto P(t, s)$ is a normalized probability distribution. The most likely analysis t_s of the sentence s is the analysis which maximizes the probability $P(t, s)$, i.e.

$$t_s = \arg\max_t P(t, s).$$

The distribution P can consist of a number of component distributions P_i, each giving probability $P_i(s, t)$ for sentence s and analysis t. The component probabilities are combined using some function $F : [0, 1]^n \to [0, 1]$ to obtain

$$P(s, t) = F\big(P_i(t, s), \ ..., \ P_n(t, s)\big).$$

The function F should be chosen in such a way that P is nonnegative and satisfies

$$\sum_{t \in T} P(t, s) = 1$$

for each sentence s.

Often a convex linear function F is used to combine estimates given by the component models. In such a case the model P is called a *linear interpolation* of the models P_i.

4 A Probabilistic First Order Model

In this section we describe the idea behind our POS tagger. We use a bigram model for POS tagging. Thus the probability of a given tagging of a sentence is estimated using analyzes of word pairs.

Since we make use of extensive training material, we may include lemmas in bigrams. Although the training material is extensive, the tagger will still encounter bigrams which did not occur in the training material or only occurred once or twice. In such cases we want to use unigram probabilities for estimating the best POS analysis. Hence we weight all analyzes using probabilities given by both the unigram and bigram models, but weight bigram probabilities heavily while only giving unigram probabilities a small weight. Hence unigram probabilities become significant only when bigram probabilities are very close to each other.

4.1 The Unigram Model

The unigram model emits plain unigram probabilities $p_u(t, s_x)$ for analyzes t given a word form s_x (we use the index x to signify that $p_u(t, s_x)$ is independent of the context of the word form s_x). Unigram probabilities are readily computed from training material. The probability of the analysis $t = (t_1 \ ... \ t_n)$ given the sentence $s = (s_1 \ ... \ s_n)$ assigned by the unigram model is

$$P_u(t, s) = \prod_{i=1}^{n} p_u(t_i, s_i).$$

In practice it is not possible to train the unigram model for all possible word forms in highly inflecting languages with productive compounding mechanism such as Finnish or Turkish. Instead the probabilities for analyzes given a word form need to be estimated using probabilities for words with similar suffixes. For instance, if the word form *foresaw* was not observed during training, we can give it a similar distribution of analyzes as the word *saw* receives, since *saw* shares a three-letter suffix with *foresaw*.

In practice such estimation relying on analogy is accomplished by a so called POS guesser, which seeks words with maximally long suffixes in common with an unknown word. It then assigns probabilities for POS analyzes of the unknown word on basis of the analyzes of the known words. [Linden 2009a] shows how a guesser can be integrated with a weighted lexicon in a consistent way.

4.2 The Bigram Models

We use two bigram models Q_o and Q_e giving probabilities for bigrams starting at even and odd positions in the sentence. The estimates are built using plain bigram probabilities for tagging a word-pair s_1 and s_2 with analyzes t_1 and t_2 respectively[1]. These probabilities $p_b(t_1, s_1, t_2, s_2)$ are easily computed from a training corpus.

For an analysis $t = t_1 \ ... \ t_{2k}$ and a sentence $s = s_1 \ ... \ s_{2k}$ of even length $2k$, the models Q_o and Q_e give bigram scores

$$Q_o(t, s) = \prod_{i=1}^{k} p_b(t_{2i-1}, s_{2i-1}, t_{2i}, s_{2i}),\ Q_e(t, s) = \prod_{i=1}^{k-1} p_b(t_{2i}, s_{2i}, t_{2i+1}, s_{2i+1})$$

For an analysis $t = t_1 \ ... \ t_{2k+1}$ and a sentence $s = s_1 \ ... \ s_{2k+1}$ of odd length $2k + 1$, the models Q_o and Q_e give bigram scores

$$Q_o(t, s) = \prod_{i=1}^{k} p_b(t_{2i-1}, s_{2i-1}, t_{2i}, s_{2i}),\ Q_e(t, s) = \prod_{i=1}^{k} p_b(t_{2i}, s_{2i}, t_{2i+1}, s_{2i+1})$$

[1] In literature, it is often suggested that one should instead compute probabilities of word form bigrams given POS analysis bigrams. We cannot do this, since we include lemmas in POS analyzes. This makes the probability of a word form given a POS analysis either 0 or 1 since most analyzes only have one realization as a word form.

4.3 Combining the Unigram and Bigram Models

The standard way of forming a model from P_u, Q_o and Q_e would be to use linear interpolation. We do not want to do this, since we aim to convert probabilities into penalty weights in the tropical semiring using the mapping $p \mapsto -\log p$, which is not compatible with sums. Instead we take a weighted product of powers of the component probabilities. Hence we get a model

$$P(t,s) = P_u(t,s)^{w_u} Q_o(t,s)^{w_o} Q_e(t,s)^{w_e}$$

where w_u, w_e and w_o are parameters, which need to be estimated.

If each of the models P_u, Q_e and Q_o agree on the probability p of an analysis t given a sentence s, we want P to give the same probability. This is accomplished exactly when $w_u + w_e + w_o = 1$. There does not seem to be any reason to prefer either of the models Q_e or Q_o, which makes it plausible to assume that $w_e = w_o$. Hence an implementation of the model only requires estimating two non-negative parameters: the unigram parameter w_u and the bigram parameter w_b. They should satisfy $w_u + 2w_b = 1$.

It is possible that $P(t,s)$ will not be a normalized distribution when s is kept fixed, but it can easily be normalized by scaling linearly with factor $\Sigma_t P(t,s)$. For the present implementation, it is not crucial that P is normalized.

5 Implementing the Statistical Model Using Weighted Finite-State Transducers

We describe the implementation of the POS tagger model using weighted finite-state transducers (WFSTs). We implement each of the components of the statistical model as a WFST, which are trained using corpus data.

In order to speed up computations and prevent roundoff errors, we convert probabilities p, given by the models, into penalty weights in the tropical semiring using the transformation $p \mapsto -\log p$. In the tropical semiring the product of probabilities pq translates to the sum of corresponding penalty weights $-\log p + -\log q$. The kth power of the probability p, namely p^k, translates to a scaling of its weight $-k \log p$. These observations follow from familiar algebraic rules for logarithms.

In our system, tagging of sentences is performed in three stages using four different WFSTs. The first two WFSTs, a weighted lexicon and a guesser for unknown words, implement a unigram model. They produce weighted suggestions for analyzes of individual word forms. The latter two WFSTs re-score the suggestions using bigram probabilities. The weights $-\log p$ given by the unigram model and the bigram model are scaled by multiplying with a constant in order to prefer analyzes which are strong bigrams. The scaled weights $-k \log p$ are then added to give the total scoring of the input sentence. This corresponds to multiplying the powers p^k of the corresponding probabilities.

In the first stage we use a weighted lexicon, which gives the five best analyzes for each known word form. In initial tests, the correct tagging for a known word

could be found among the five best analyzes in over 99% of tagged word forms, so we get sufficient coverage while reducing computational complexity.

For an unknown word x, we use a guesser which estimates the probability of analyzes using the probabilities for analyzes of known words. We find the set of known word forms W, whose words share the longest possible suffix with the word form x. We then determine the five best analyzes for the unknown word form x by finding the five best analyzes for words in the set W.

For each word s_i in a sentence $s = s_1 \ldots s_n$, we form a WFST W_i which is a disjunction of its five best analyzes $t_1 \ldots t_5$ according to the weights $w(s_i, t_i)$ given by the unigram model. In case there are less than five analyzes for a word, we take as many as there are. We then compute a weighted concatenation W_s of the individual WFSTs W_i. The transducer W_s is the disjunction of all POS analyzes of the sentence s, where each word receives one of its best five analyzes given by the unigram model.

To re-score the analysis suggestions given by the lexicon and the guesser, we use two WFSTs whose combined effect gives the bigram weighting for the sentence. One of the model scores bigrams starting at even positions in the sentence and the other one scores bigrams starting at odd positions. Thus we give a score for all bigrams in the sentence without having to compute a WFST equivalent to the intersection of the models which might be quite large.

Using weighted intersecting composition [Silfverberg and Lindén 2009] we simultaneously apply both bigram scoring WFSTs to the sentence WFST W_s. The POS analysis of the sentence s is the best path of the result of the composition.

The WFSTs and algorithms for parsing were implemented using the Helsinki Finite-State Technology (HFST) interface [Linden et al. 2009].

We now describe the lexicon, guesser and the bigram WFSTs in more detail.

5.1 The Weighted Lexicon

Using a tagged corpus, we form a weighted lexicon L which re-writes word forms to their lemmas and analyzes. POS analyzes for a word form s_i are weighted according to their frequencies, which are transformed into tropical weights.

In order to estimate the weights for words which were not seen in the training corpus, we construct a guesser. For an unknown word, the guesser will try to construct a series of analyzes relying on information about the analyzes of known similar words.

Figure 1 shows an example guesser, which can be constructed from a reversed weighted lexicon. Guessing begins at the end of the word. We allow guessing at a particular analysis for a word only if the word has a suffix agreeing with the analysis. See [Linden 2009a] for more information on guessers.

5.2 The Bigram Models

To re-score analyzes given by the unigram model, we use two WFSTs whose combination serves as a bigram model. The first one, B_e, scores each known word form/analysis bigram s_{2k}, s_{2k+1} and t_1, t_2 in the sentence starting at an

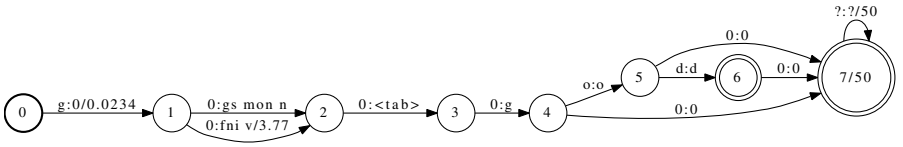


Fig. 1. Guesser constructed from a weighted lexicon. Guessing starts at the end of a word. Skipping letters gives a high penalty and analyzes, where equally many letters are skipped, are weighted according to the frequency of the analyzes.

even position $2k$ according to the maximum likelihood estimate of the tag bigram t_1t_2 w.r.t. the word form bigram $s_{2k}s_{2k+1}$. The WFST B_o is similar to B_e except it weights bigrams starting at odd positions $s_{2k-1}s_{2k}$.

Given a word form pair s_1, s_2, we compute the probability $P(t_1, s_1, t_2, s_2)$ for each POS analysis pair t_1, t_2. These sum to 1 when w_1 and w_2 remain fixed. Then we form a transducer B, whose paths transform word form pairs s_1s_2 into analysis pairs t_1t_2 with weight $-\log P(t_1, s_1, t_2, s_2)$. Lastly we disjunct B with a default bigram, which transforms arbitrary word form sequences to arbitrary analyzes with a penalty weight, which is greater than the penalty received by all other transformations.

In addition to the model B, we also compute a general word model W, which transforms an arbitrary sequence of symbols into an arbitrary lemma and an analysis. The word model W is used to skip words at the beginning and end of sentences.

From the transducers above, we form the models B_e and B_o using weighted finite state operations

$$B_e = WB^*W^{\{0,1\}} \text{ and } B_o = B^*W^{\{0,1\}}.$$

Here $W^{\{0,1\}}$ signifies an optional instance of W.

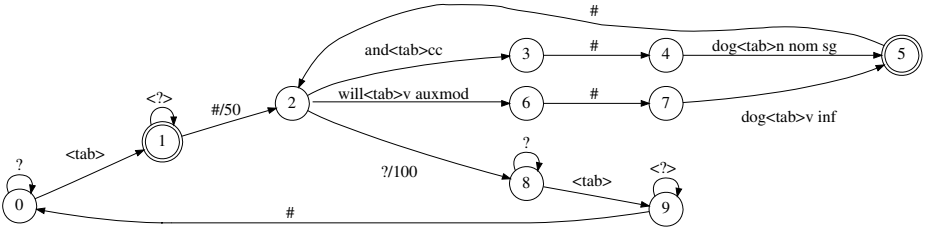


Fig. 2. A small example of an even bigram model B_e. ? signifies an arbitrary symbol and <?> signifies an arbitrary POS analysis symbol.

5.3 Parsing Using Weighted Intersecting Composition

In our system, parsing a sentence S is in principle equivalent to finding the best path of the transducer

$$(S \circ L) \circ (B_e \cap B_o).$$

Since the intersection of B_o and B_e could become prohibitively large, we instead use intersecting composition [Silfverberg and Lindén 2009] to simulate the intersection of B_e and B_o during composition with the unigram tagged sentence $S \circ L$.

Intersecting composition is an operation first used in compiling two-level grammars [Karttunen 1994]. We use a weighted version of the operation.

After the intersecting composition, we extract the best path from the resulting transducer. This is the tagged sentence.

6 Data

In this section we describe the data used for testing and training the POS tagger.

For testing and training, we used the Europarl parallel corpus [Koehn 2005].

The Europarl parallel corpus is a collection of proceedings of the European Parliament in eleven European languages. The corpus has markup to identify speaker and some html-markup, which we removed to produce a file in raw text format. We used the Finnish, English and Swedish corpora. Since the training and testing materials are the same for all three languages, the results we obtain for the different languages are comparable.

We parsed the Europarl corpora using Connexor functional dependency parsers fi-fdg for Finnish, sv-fdg for Swedish and en-fdg for English [Järvinen et al. 2004]. From the parses of the corpora we extracted word forms, lemmas and POS tags. For training and testing, we preserved the original tokenization of the fdg-parsers and removed *prop* tags marking proper nouns, *abbr* tags marking abbreviations and *heur* tags marking guesses made by the fdg-parser. The tag sequence counts in table 1 represent the number of tag sequences after *abbr*, *prop* and *heur* tags were removed.

Table 1. Some figures describing the test and training material for the POS tagger

Language	Syntactic tokens	Sentences	POS tag sequences
English	43 million	1 million	122
Finnish	25 million	1 million	2194
Swedish	38 million	1 million	243

Table 1 describes the data used in training and testing the POS tagger. We see that the fi-fdg parser for Finnish emitted more than ten times as many tag sequences as sv-fdg for Swedish or en-fdg for English. The en-fdg parse emitted clearly fewest tag sequences.

7 Training the Model

We now describe training the model, which consists of two phases. In the first phase we build the weighted lexicon and guesser and the bigram models. In the second phase we estimate experimentally coefficients w_u and w_b, which maximize the accuracy of the interpolated model

$$P(t,s) = P_u(t,s)^{w_u} Q_o(t,s)^{w_b} Q_e(t,s)^{w_b}$$

Using a small material covering 1000 syntactic tokens, we estimated $w_u = 0.1$ and $w_b = 0.45$. This shows that it is beneficial to weight the bigram model heavily, which seems natural, since bigrams provide more information than unigrams.

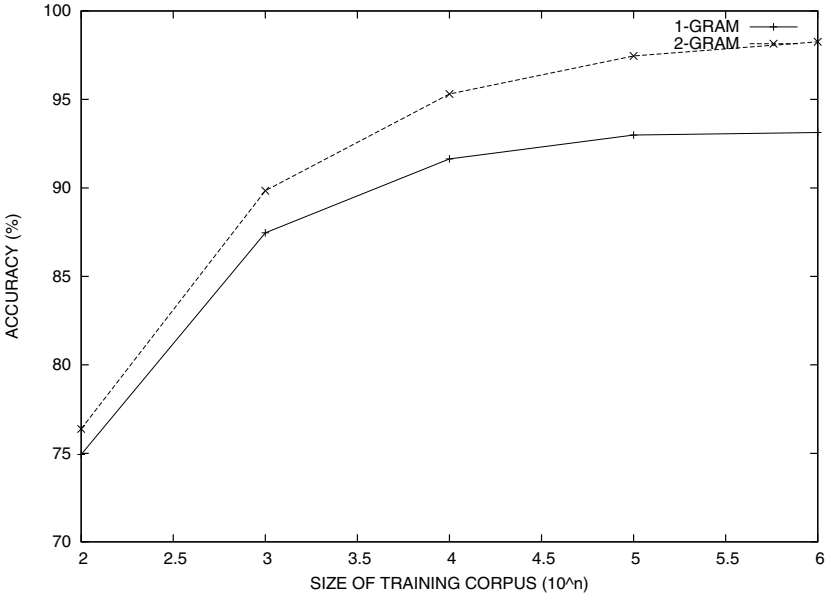


Fig. 3. The accuracy for the English POS tagger as a function of the size of training data. We used between 10^2 and 10^6 sentences for training. The lower curve displays the accuracy using only the unigram model, whilst the upper curve displays the accuracy of the combined unigram and bigram model.

Figure 3 shows learning curves for the English language POS tagger using 10^2 to 10^6 sentences for training. The lower curve displays accuracies for the unigram model and the upper curve shows the accuracy for the combined unigram and bigram model. For the unigram model, we can see that little improvement is obtained by increasing the training data from 10^4 sentences. In contrast, there is significant improvement ($\approx 0.82\%$) for the bigram model even when we move from 10^5 to 10^6 sentences.

8 Evaluation

We describe the methods we used to evaluate the POS tagger and the results we got.

We used ten-fold cross-validation to evaluate the POS tagger, that is we split the training material in ten equally sized parts and used nine parts for training the model and the remaining part for testing. Varying the tenth used for testing we trained ten POS taggers for each language.

For each of the languages we trained two sets of taggers. One set used only unigram probabilities for assigning POS tags. The other used both unigram and bigram probabilities. We may consider the unigram taggers as a baseline.

For each tree languages, we computed the average and standard deviation of the accuracy of the unigram and bigram taggers. In addition we computed the Wilcoxon matched-pairs signed-ranks test for the bigram and unigram accuracies in all three languages. The test does not assume that the data is normally distributed (unlike the paired t-test). The results of our tests can be seen in table 2.

Table 2. Average accuracies and standard deviations for POS taggers in Finnish, English and Swedish. The sixth column shows the improvement, which results for adding the bigram model. In the seventh column, we show the results of the Wilcoxon matched-pairs signed-ranks test between unigram and bigram accuracies.

Language	Unigram Acc.	σ	Bigram Acc.	σ	Diff.	Conf.
English	93.10%	0.09	98.29%	0.01	5.19%	$\geq 99.8\%$
Finnish	94.38%	0.07	96.63%	0.03	2.25%	$\geq 99.8\%$
Swedish	94.12%	0.20	97.31%	0.11	3.19%	$\geq 99.6\%$

9 Discussion and Future Work

It is interesting to see that a bigram tagger can perform equally well or better than trigram taggers at least on certain text genres. The mean accuracy 98.29%, we obtained for tagging the English Europarl corpus is exceptionally high (for example [Shen et al. 2007], report a 97.33% accuracy on tagging the Penn Tree Bank). The improvement of 5.19 percentage points from the unigram model to the combined unigram and bigram model is also impressive. There is also a clear improvement for Finnish and Swedish, when the bigram model is used in tagging and accuracy for these languages is also high. We had problems finding accuracies figures for statistical taggers of Finnish, but for Swedish [Megyesi 2001] reports accuracies between 94% and 96%, which means that we get state-of-the-art accuracy for Swedish.

Of course the Europarl corpus is probably more homogeneous than the Penn Tree Bank or the Brown Corpus, both of which include texts from a variety of genres. Furthermore tagging is easier because the en-fdg parser only emits 122 different POS analyzes. Still, Europarl texts represent an important genre,

because the EU is constantly producing written materials, which need to be translated into all official languages of the union.

The accuracy for Finnish shows less improvement than English and Swedish. We believe this is a result of the fact that Finnish words carry a lot of information but the bonds between words in sentences may be quite weak. This conclusion is supported by the fact that unigram accuracy for Finnish is best of all three languages.

We do not believe, that using trigram statistics would bring much improvement for Finnish. Instead we would like to write a set of linguistic rules which would cover most typically occurring tagging errors. Especially we would like to try out constraints, which would mark certain analyzes as illegal in some contexts. Such negative information is hard to learn using statistical methods. Still, it may be very useful, so it could be provided by hand-crafted rules.

Clearly our figures for accuracy need to be considered in relation to the tagging accuracy of the fdg parsers. We did not succeed in finding a study on the POS tagging accuracy of the fdg parsers. Instead we examined the POS tagging for one word per twenty thousand in the first tenth of the Europarl corpora for Finnish, English and Swedish. This amounted to 131 examined words for Finnish, 219 examined words for English and 191 examined words for Swedish. According to these tests, the POS tagging accuracy of the fdg parsers for Finnish is 95.4%, for English it is 97.3% and for Swedish it is 97.5%.

10 Conclusions

We introduced a model for a statistical POS tagger using bigram statistics with lemmas included. We showed how the tagger can be implemented using WFSTs. We also demonstrated a new way to factor a first order model into a model tagging bigrams at even positions in the sentence and another model tagging bigrams at odd positions.

In order to test our model, we implemented POS taggers for Finnish, English and Swedish, training them and evaluating them using Europarl corpora in the respective languages and Connexor fdg parsers.

We obtained a clear, statistically significant, improvement for all three languages when compared to the baseline unigram tagger. At least for English and Swedish, we obtain state-of-the-art accuracy.

Acknowledgements. We thank the anonymous referees. We also want thank our colleagues in the Hfst team. The first author is funded by Langnet Graduate School for Language Studies.

References

[Brants 2000] Brants, T.: TnT – A Statistical Part-of-Speech Tagger. In: ANLP - 2000 (2000)

[Church 1999] Church, K.: A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In: Proceedings of the Second Conference on Applied Natural Language Processing (1988)

[Collins 2002] Collins, M.: Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In: EMNLP (2002)

[Cutting 1992] Cutting, D., Kupiec, J., Pedersen, J., Sibun, P.: A Practical Part-of-Speech Tagger. In: Proceedings of the Third Conference on Applied Natural Language Processing (1992)

[Järvinen et al. 2004] Järvinen, T., Laari, M., Lahtinen, T., Paajanen, S., Paljakka, P., Soininen, M., Tapanainen, P.: Robust Language Analysis Components for Practical Applications. In: Gambäck, B., Jokinen, K. (eds.) Robust and Adaptive Information Processing for Mobile Speech Interfaces (2004)

[Karttunen 1994] Karttunen, L.: Constructing Lexical Transducers. In: COLING 1994, pp. 406–411 (1994)

[Koehn 2005] Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. In: Machine Translation Summit X, Phuket, Thailand, pp. 79–86 (2005)

[Koskenniemi 1990] Koskenniemi, K.: Finite-state parsing and disambiguation. In: 13th COLING (1990)

[Lafferty et al. 2001] Lafferty, J., MacCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML 2001 (2001)

[Linden et al. 2009] Lindén, K., Silfverberg, M., Pirinen, T.: Hfst Tools for Morphology – an Efficient Open-Source Package for Construction of Morphological Analyzers. In: Mahlow, C., Piotrowski, M. (eds.) SFCM 2009. LNCS, vol. 41, pp. 28–47. Springer, Heidelberg (2009)

[Linden 2009a] Lindén, K.: Entry Generation by Analogy Encoding New Words for Morphological Lexicons. NEJLT, vol. 1 (2009)

[Linden 2009b] Lindén, K.: Guessers for Finite-State Transducer Lexicons. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 158–169. Springer, Heidelberg (2009)

[Megyesi 2001] Megyesi, B.: Comparing data-driven learning algorithms for POS tagging of Swedish. In: EMNLP 2001 (2001)

[Mohri and Riley 2002] Mohri, M., Riley, M.: An Efficient Algorithm for the n-Best-Strings Problem. In: ICSLP 2002 (2002)

[Mikheev 1997] Mikheev, A.: Automatic Rule Induction for Unknown-Word Guessing. In: CL, vol. 23 (1997)

[Shen et al. 2007] Shen, L., Satta, G., Joshi, A.: Guided Learning for Bidirectional Sequence Classification. In: ACL 2007 (2007)

[Silfverberg and Lindén 2009] Silfverberg, M., Lindén, K.: Conflict Resolution Using Weighted Rules in HFST-TwolC. In: NODALIDA 2009 (2009)

[Thede and Harper 1999] Thede, S., Harper, M.: A Second-Order Hidden Markov Model for Part-of-Speech Tagging. In: 37th ACL (1999)

Automated Email Answering by Text Pattern Matching

Eriks Sneiders

Department of Computer and Systems Sciences,
Stockholm University,
Forum 100, SE-16440 Kista, Sweden
eriks@dsv.su.se

Abstract. Answering email by standard answers is a common practice at contact centers. Our research assists this process by creating reply messages that contain one or several standard answers. Our standard answers are linked to representative text patterns that match incoming messages. The system works in three languages. The performance was evaluated on two email sets; the main advantage of our email answering technique is good correctness of the delivered replies.

Keywords: Automated email answering, automatic email response, text message answering, question answering, text patterns.

1 Introduction

It is not unusual that an email flow to a contact center (aka customer care center, customer service) contains frequently reoccurring inquiries, therefore agents who communicate with customers use predefined response templates as draft answers. Finding a predefined answer, if it exists, is a task that a computer can do. Answering a generic question (e.g., "Can I pay with my Visa card?") would be easy. More specific requests (e.g., "Please update my address in your customer database…") are less trivial. Fortunately, many companies have a web-based self-service system where a customer logs in and interacts with the system without mediation of a contact center agent. Hence, an automated answer can advise using the self-service system, where appropriate.

Katakis et al. [1] present a good introduction to email management techniques and their application domains. Most research has been done assuming personal use of email. Automated message answering at contact centers has raised less interest.

Most email answering systems pursue the text classification approach. A typical system perceives a message as a bag of words represented by a term vector with tf-idf weights. Normally the words are stemmed and stop-words removed. Further, a typical system is trained on sample documents in predefined classes. The most popular learning algorithms are Support Vector Machine and Naïve Bayes [2][3][4]; Ripper and K Nearest Neighbors have also been used [3]. After the training phase, a new message is placed into one of the predefined classes, e.g., assigned a predefined answer or a small set of candidate answers.

Malik et al. [5] map training email messages to standard answers by identifying questions that contain key phrases of length up to 3 words. After the training, these questions are matched to questions in query emails.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 381–392, 2010.

Weng and Liu [6] assign a set of representative concepts with weighted terms to each class of messages. When a new message arrives, its weight is calculated with respect to each concept set considering the terms in the message and their weights.

Very few email answering systems do text generation. Marom and Zukerman [7] create new response texts by selecting the most representative sentences from previous responses to messages similar to the new incoming message. Kosseim at al. [8] follow the tradition of Information Extraction and operate a number of templates for capturing intention, concepts, named entities, and relations. When a new message arrives, the system fills the templates, performs semantic validation, and generates the answer.

This paper introduces an email answering approach that has been used for sending replies without any human intervention as well as generating draft replies for contact center agents. The system operates a database of standard answers and text patterns, linked to these answers, that record expected wordings in messages to be answered.

2 Pattern-Based Email Answering

This research stems from an earlier work in automated FAQ answering in a restricted domain: the system could answer about 70% of the queries, 9 out of 10 answers correct [9]. A natural step forward is to adapt the technique to answer larger pieces of text, such as email messages.

2.1 Question Templates

The email answering system has a few standard answers that respond to the most frequent inquiries. Each standard answer has a title that summarizes it in one sentence, which helps to avoid confusion while reading the automatic reply if the answer does not quite correspond to the original message. Furthermore, each standard answer is linked to a number of question templates that record the expected text patterns of the future inquiries to be answered by this standard answer, Fig. 1.

The syntax of our text patterns resembles that of regular expressions; the text patterns are less rigorous though. Each question template contains two patterns. The required pattern matches a piece of message text if the message fits this standard answer. The forbidden pattern must not match the message; it detects details that disqualify the answer. Please observe that the question templates are created before the actual email answering starts.

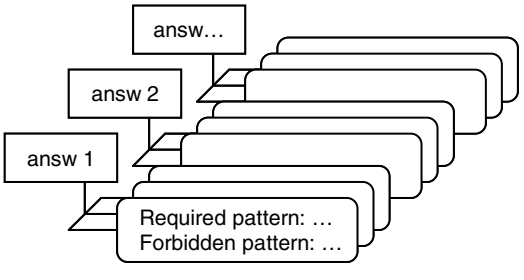


Fig. 1. Standard answers with their question templates

2.2 Steps of the Answering Process

During the email answering process, a new incoming message is split into paragraphs, each paragraph into sentences, each sentence into terms – words, numbers, and email addresses. Then the words are spell-checked (see Section 3); all spelling alternatives are equal.

After the message has been preprocessed, the system matches each question template, one by one, to each paragraph of the message, one by one. The matching is case-insensitive. If the forbidden pattern matches at least one paragraph, the system drops the current question template and starts all over again with the next question template. If the required pattern matches at least one paragraph, and the forbidden pattern does not match any, the system accepts the standard answer linked to the current question template.

One incoming message may contain several inquiries and match several question templates, and therefore be answered by several standard answers included in one reply message. The length of the message is not limited. Nonetheless, if the message matches too many standard answers, we would rather not answer it automatically.

2.3 Components of a Text Pattern

A text pattern, which matches a piece of a text message, is a collection of lexical items and rules that state how the system matches those items to terms in a paragraph of a query message. A text pattern is written in plain text following a simple syntax.

There are three types of basic lexical items: words and word stems (e.g., customer* or %paper), numbers (e.g., 29), and entity slots. An entity slot represents any item of a given concept. Currently we have entity slots for numbers and email addresses, written as <number> and <email>. The diversity of possible entity slots is limited only by the systems ability to recognize a piece of text as a concept, and isolate that piece during email text tokenization.

Lexical items are organized into synonym sets. For example, ford volvo ; car vehicle* ; repair* are three synonym sets, where at least one synonym per set, in every set, should match any term in the query paragraph.

A phrase defines the order of and the distance between synonym sets. A phrase itself is a synonym in another synonym set. Phrases can be embedded. A phrase matches terms within the boundaries of one sentence. A phrase defines three neighborhood options for matching terms:

- adjacent, e.g., [bright; light] matches "bright light" only;
- distant, e.g., [bright # light] allows unlimited number of words between "bright" and "light" within one sentence;
- optional, e. g., [bright : day ; light] matches both "bright light" and "bright day light".

An empty synonym set matches everything, therefore [bright :::; light] matches “bright” and “light” with 0-3 any lexical items in between.

Phrases match compound words. For example, [new*; paper*] matches “new paper” and “newspapers”. Because compound words are popular in some languages such as Swedish, we have special syntax for them, e.g., news%paper* is equal to [news* ; paper*].

We define reoccurring pieces of text patterns as substitutes and reuse them. For example, we define ford volvo as $car_name and use it in $car_name ; car vehicle* ; repair*.

In order to minimize words stem ambiguity, the stems are required to have at least five letters before the asterisk. In the syntax for compound words, a component must be at least four letters long, or at least two if there is another component at least five letters long. This has proved a sufficient trade-off between the ambiguity of word stems and their ability to represent concepts.

2.4 Test Beds

Before we discuss real-life examples of the text patterns, let us introduce our test beds. The email answering system was implemented at two contact centers. The first one was an insurance company that employed fully automated email answering with 11 standard answers in Swedish. The system scanned through all incoming messages. If it could answer the message, it sent a reply to the "from" and "cc" addresses of the original message. The reply informed its reader that it was computer-created and contained simple instructions how to reach a human agent if necessary. Messages that could not be answered were passed to a human agent.

The second contact center worked for a telecom service provider. It used 4 standard answers in Latvian. Here, email answering was not fully automated; the system created draft reply messages for the agents of the contact center.

2.5 Flexibility of a Text Pattern

The following example illustrates what a text pattern may look like:

```
[ $2_4_hjul #
    [vad ::; $kosta :::; $försäkring $2_4_hjulförsäkring $trafikförsäkring $hel_halv_försäkring]
    [vad ::; $försäkring $2_4_hjulförsäkring $hel_halv_försäkring :::::; $kosta $pris2]
    [$försäkring $2_4_hjulförsäkring $trafikförsäkring $hel_halv_försäkring # vad ::::::;
        $kosta $pris2]
]
```

A text pattern is more than a loose regular expression; it embodies a question-specific synonym dictionary and a quasi-ontology. Furthermore, a combination of words and matching rules is delicate, even small changes may influence the accuracy of detecting relevant texts. Therefore, at the current stage of our research, management of the text patterns requires a manual effort.

In order to get an impression about the potential size of a text pattern, let us see how many terms in a piece of query text (i.e., a paragraph) one pattern can match, taking the insurance case (see Section 2.4) as an example. The 11 standard answers were linked to 161 required text patterns. In average, a text pattern matched 5 terms in a paragraph if the system delivered an answer. Potentially the text patterns could match between 3 and 40 terms in a paragraph.

Fig. 2 shows the largest (top curve) and smallest (bottom curve) number of terms in a query paragraph that a text pattern could possibly match, for each of the 161 patterns lined up on the horizontal axis. On the very left side, there are text patterns that can match just over 40 query terms. In the middle of the line-up, the text patterns can match up to 15-20 query terms. The smallest text patterns on the right side can match just over 5

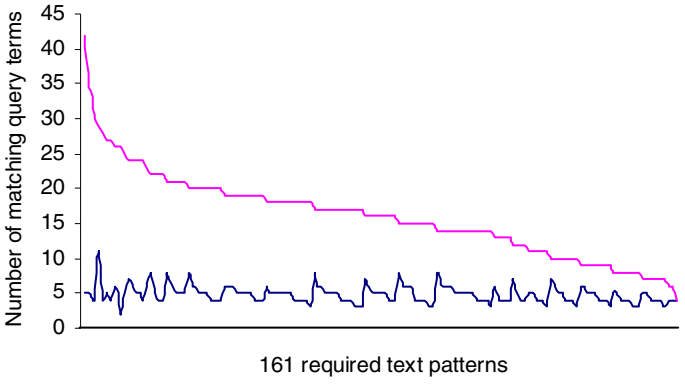


Fig. 2. Number of query terms that can match a required text pattern

terms in a query paragraph. The bottom curve oscillates like an electrocardiogram, with average 4.8 terms in a query paragraph that a text pattern must match as a minimum.

Why can one text pattern match a variable number of query terms? A synonym set may contain phrases of different length. The largest/smallest number of matching terms can be reached if the system always selects the longest/shortest synonym phrase, or a basic lexical item instead of the shortest phrase, in each synonym set. Fig. 2 suggests that a text pattern may contain a large number of parallel wordings that capture a variety of paraphrases of a meaningful statement. Furthermore, text patterns are built to match future messages, and we are uncertain what exactly these messages will look like. We are likely to compensate this uncertainty with some redundancy in the text patterns caused by guessing the future wordings. Hereby the text patterns are more complex but also more expressive than just a set of keywords and a synonym dictionary.

3 Spelling Correction

Email texts are often untidy. Tang et al. [10] inspected more than five thousand web-based newsgroup messages in English and discovered that 73.2% of them needed paragraph normalization, 85.4% needed sentence normalization, 47.1% needed upper-lower case restoration, and 7.4% of the messages contained misspellings. Dalianis [11] inspected spelling in another context of Internet-based communication and found out that about 10% of the search queries submitted to a search engine of the Swedish tax authority were misspelled. We did not count misspelled messages processed by our system, yet we know that without spelling alterations our text-pattern matching would fail to identify many pertinent messages because the matching rules, embodied in these patterns, are strict, and even one letter wrong would result in a no-match.

Furthermore, our system faces three challenges. First of all, spelling alternatives are sought among words, word stems, and phrases from the text patterns rather than in off-the-shelf tools.

The second challenge is multiple languages. The system works with texts in English, Swedish (Germanic/Indo-European languages), and Latvian (Baltic/Indo-European language).

The third challenge is substitution of language-specific character sets with the ISO-8859-1 or ASCII character sets present on virtually all computers. Of the three languages that our system works with, Latvian emails are most exposed to character set substitution. The Latvian alphabet has 33 letters, of which 11 are not included in the ISO-8859-1 character set and may get replaced with English letters. For example, "ā" becomes "a" or "aa", "ķ" becomes "k" or "kj" or "kk", etc. There are no rules, everything goes as long as people grasp the text. In this case we deal with a deliberately altered syntax, a pidgin language, rather than misspellings. The cure, however, is the same – spelling correction.

Our spelling correction approach has three building blocks: (i) detection of correctly spelled words, (ii) spelling modification with respect to standalone words and stems from the text patterns, and (iii) spelling modification with respect to phrases from the text patterns. The same algorithms are applied to texts in all three languages, except the system does not detect correct spelling in Latvian emails.

Detection of correct spelling and common misspellings. Correctly spelled words are detected by matching them to words, word stems, and phrases from the text patterns, as well as word lists available in public domain or donated by colleagues. Likewise, the system matches query words to a list of common misspellings and finds correct spelling for commonly misspelled words.

Detection of correctly spelled and frequently misspelled words is not performed to messages in Latvian because of frequent occurrence of pidgin language in them, where a misspelling of one word may be a homograph of correctly spelled another word. For example, "kāpa" and "kapa" are correctly spelled different words, yet "kapa" may be inappropriately used instead of "kāpa". Thus spelling correction is closely related to the problem of word sense disambiguation [12], which lies outside the scope of this paper. Our system assumes that every word in Latvian emails is potentially misspelled.

Words and word stems as spelling alternatives. About 80% of misspellings fall into four categories: one letter too many, one letter missing, a letter replaced by another one, or adjacent letters are transposed [13]. Our spelling correction is based on detecting these four typical mistakes. We tested also basic Soundex for English; it generated too many irrelevant options. There exist more advanced spelling error correction models, such as [14]. Still, advances in spelling correction lie outside the scope of this paper.

Our detection of replaced letters considers also sequences of letters, and looks for:

- phonetic similarity, e.g., "f" ~ "ph", "ea" ~ "ee", "k" ~ "ck", "s" ~ "c", etc.
- visual similarity, e.g., "ä" ~ "a", "š" ~ "s", etc.
- conventions in pidgin languages, e.g., "ä" ~ "ae", "š" ~ "sh" ~ "sch", etc.

If a standalone word stem (not a part of a phrase) is considered as a spelling alternative, the corrected word must be no more than three letters longer than the stem in order to hinder replacement of a compound word with the stem of its first component.

Phrases as spelling alternatives. Phrases from the text patterns, simple ones without other embedded phrases, are the main spelling alternatives for misspelled compound words. The system tests a phrase as a spelling alternative of a query word according to the following principle:

- the system applies synonyms from a synonym set to a query word
- in case if a stem fits the query word, the system takes the remainder of the query word, left behind the fitting stem, as a new query word
    - the system takes the next synonym set of the phrase as the current synonym set and repeatedly applies it to the new (remainder) query word, dropping the first letter of the new query word after each iteration in order to detect the beginning of the next component of a compound word
- if the remainder does not fit the current synonym set, the system takes the next query word and applies the current synonym set

Let us consider an example and test [fågel* fåglar* ; bur] as a spelling alternative of "faogelsbuur". First, the system discovers that fågel* fits "faogel". Then it takes the remainder "sbuur" and repeatedly applies bur, after each iteration dropping the first letter of the remainder, until bur fits "buur". Let us now try "faoglarnas buur". The system discovers that fåglar* fits "faoglar", that bur does not fit the remainder "nas", but it does fit the next query word "buur".

Order of the rules. The system conducts spelling correction in the following manner:

1. Correctly spelled words are detected, commonly misspelled words are corrected, for texts in English and Swedish.
2. Phrases from the text patterns are tested as spelling alternatives, longer phrases first. Corrected compound words are not further processed applying shorter phrases and standalone word stems.
3. Standalone words and word stems from the text patterns are tested as spelling alternatives.

When a word or a word stem is considered as a spelling alternative, the four typical mistakes are tested in the following order: (1) replaced letters, (2) adjacent letters transposed, (3) one letter too much or missing, tested simultaneously.

4 Performance Measurements

Before the email answering system can start operating, it is "trained" to recognize email texts that fit a given standard answer. We say "trained", in quotes, because this is not training as understood in machine learning. We use some text filtering, clustering, and aggregation tools in order to group messages. Then the messages are analyzed and text patterns created. Today, the text patterns are crafted manually; increased automation of this process is further research and lies outside the scope of this paper.

"Training" messages. Section 2.4 introduced our test beds – one contact center for a Swedish insurance company, another one for a Latvian telecom service provider.

In the insurance case, we had access to 5148 messages before the performance test; many of these messages had been analyzed in order to manually create and adjust the text patterns.

In the telecom case, we had access to two sets of messages. Initially we had 4782 messages, of which 706 messages corresponded to the 4 standard answers that the system included in its automated replies. During the operation of the system, but before the performance test, we acquired 3768 more messages that were analyzed in order to manually adjust the text patterns.

Because the systems were running in production settings, the "training" phase was the entire period the systems had been in operation. While doing the "training", we paid more attention to correctness of the replies rather than recall. We rather process a message manually than increase the risk of an incorrect reply.

Test messages. The data for the performance measurements came from the systems' logs. We had no prior access to these messages, we could not have used them in order to "train" the systems.

In the insurance case, 3526 consecutive messages were analyzed. In the telecom case, 1314 consecutive messages were analyzed. The correspondence between query messages and their automated replies was judged by humans, third party observers.

4.1 Precision, Recall, and Correctness

We applied two evaluation criteria. The first criterion was precision and recall, calculated for each query message separately. The second criterion was correctness of the replies actually given. We distinguished between:

- a fully correct reply;
- a correct partial reply where the system did not possess answers to all the questions in the query (recall = 1);
- a correct partial reply where the system did not find all the answers (recall < 1);
- an incorrect reply (usually right concepts, wrong relations);
- a technically correct reply where the system properly identified the question or the problem statement, but the message further implied that the author could not use the standard solution proposed in the reply.

Table 1 shows precision and recall calculated for each query message separately. Let us explain the insurance case. From the 3526 inspected messages, 395 messages got recall 1, 20 messages got recall 0.5, 179 messages got recall 0. In total, 594 messages (395+20+179) had relevant answers in the system's database and some recall value. The average recall – the sum of all recall values divided by the number of messages that had any recall value – was 0.682.

Precision values where calculated only for the query messages with a non-zero recall (precision is not defined otherwise), which was 415 messages (395+20). Most messages – 404 – got precision value 1. The average precision was 0.987.

Table 2 shows correctness of the replies, i.e., how many replies actually delivered were somewhat correct or incorrect. The messages sent to manual processing do not show up in this table.

Table 1. Precision and recall of the automatic replies

Precision value	Num queries with the precision value	Recall value	Num queries with the recall value
Insurance case			
1	404	1	395
0.67	1	0.5	20
0.5	9	0	179
0.33	1		
Average: 0.987 (for 415 queries)	Total: 415	Average: 0.682 (for 594 queries)	Total: 594
Telecom case			
1	119	1	120
0.5	5	0.5	4
		0	37
Average: 0.98 (for 124 queries)	Total: 124	Average: 0.758 (for 161 query)	Total: 161

Table 2. Correctness of the automatic replies

Fully correct	Technically correct	Partial, recall=1	Partial, recall<1	Incorrect	Total
Insurance case					
371	28	24	20	41	484
76.65%	5.79%	4.96%	4.13%	8.47%	100%
Telecom case					
111	6	9	4	16	146
76.03%	4.11%	6.16%	2.74%	10.96%	100%

A sharp-eyed reader has probably noticed that numbers in Table 1 and Table 2 do not match. Let us take the insurance case in order to explain these numbers. 371 fully answered plus 24+20 partially answered queries make 415 total precision queries. Incorrectly answered queries have either zero-recall or do not have any recall and precision values at all. Technically correct replies were judged as incorrect when precision and recall where calculated, because these queries should not have been answered. Therefore technically correct replies did not get any recall and precision values.

Both implementations of the system show surprisingly similar correctness and average precision percentages, despite different languages and knowledge domains. Correctness of the replies is good. From all the answered queries,

- around 76% were answered fully,
- around 85% were answered fully or partially,
- around 90% got the issues identified, fully or partially (fully correct, technically correct, partial replies).

4.2 Performance Figures in Context

The performance figures are easier to grasp if observed in the context of related systems, mentioned in the introduction, whose performance measurement methods are similar to those of ours.

Message classification. Busemann et al. [2] managed to choose the right category in 56.23% cases using Support Vector Machine; Weng and Liu [6] reached top performance, i.e., the highest F-value, at 62.77% recall and 77.52% precision by finding representative terms in query messages, and weighing these terms with respect to message classes.

Mapping a message to a standard answer. Malik et al. [5] made a human-equivalent selection of answer templates in 61% cases, human-equivalent or incomplete selection in 73.4% by first extracting a set of representative questions for each standard answer (training), and then mapping these questions to questions in query messages (test), calculating taxonomy-distance between words.

Information extraction and answer text generation. Kosseim et al. [8] delivered 66.4% correct answers by applying Information Extraction templates and performing semantic validation of the extracted information.

These figures give us an intuitive insight into viable quality of email answering. In order to make any formal claims which system performs better, we have to test the systems using the same input data and applying the same conditions.

5 Advantages and Limitations

Following are the advantages of our text pattern-based automated email answering.

The system demonstrates a good correctness of the replies and a superior ability to identify questions and problem statements relevant to standard answers, if the message is answered.

The text patterns operate in isolated narrow single-answer knowledge domains where they are self-sufficient. Therefore the system can start operation with rendering one standard answer, and the number of standard answers can gradually rise to as many as needed.

Because each text pattern is autonomous, the system can easily handle several questions in one query message and put several standard answers into one reply message.

Because the text patterns are self-sufficient, the system has a low technological threshold or barrier that impedes its deployment, i.e., the system operates without components such as part-of-speech taggers, stemmers, generic or domain ontologies. We do need some tool support to analyze training messages, however, as stated in further research.

Because of the low technological threshold, our approach is affordable for rare domains, small businesses, and small languages where open-source linguistic and knowledge-representation tools, as well as technologically advanced manpower are not readily available.

We have tested the system for English, Swedish, and Latvian, and consider it portable to most European languages.

Our approach is most advantageous in settings where correctness of the replies is crucial, where we want to maximize the end-users' experience, where a list of ten candidate answers is not an option. For example, in fully automated email answering without any human mediation. Especially advantageous our approach is for email flows with a high ratio of reoccurring inquiries, such as the email flow in [4] where 9 standard answers cover 72% of all messages.

Still, our automated email answering approach has at least two *limitations*. First, it is designed for narrow and stable domains only. It should not be considered for text classification tasks in arbitrary text collections. Second, at the current stage of our research, development of the text patterns is manual, which lessens the practical value of our technique until at least partial automation of this process is achieved.

6 Conclusions and Further Research

Our automated email answering system maps incoming messages to standard answers by matching text patterns linked to the standard answers. The technique was designed for narrow and stable domains, such as an email flow at a contact center. The main advantage of our technique is good correctness of the delivered replies. Performance evaluation on two email collections in two languages showed that about 85% of the messages could be answered fully or partially; about 90% had their questions and problem statements correctly identified. We consider the technique applicable to the majority of the European languages.

Currently we are working on performance comparison between our system and Support Vector Machine / Naïve Bayes on different datasets, which will be published separately.

The performance of the system depends on how good its text patterns are, therefore our further research focuses on the tools that help us craft these patterns. This includes: (i) learning from the past experiences, i.e., analysis of the existing email flow, and (ii) adjustment of the acquired learnings to meet future queries.

The research continues in three directions. First, we need to know what makes a good text pattern, what features we should strive to develop. Second, we need a good key phrase extraction tool that creates the foundation for new text patterns given a sample of messages. Third, we need a machine learning tool that adjusts existing text patterns to new training messages.

References

1. Katakis, I., Tsoumakas, G., Vlahavas, I.: Email Mining: Emerging Techniques for Email Management. In: Vakali, A., Pallis, G. (eds.) Web Data Management Practices: Emerging Techniques and Technologies, pp. 219–240. Idea Group Publishing, USA (2006)

2. Busemann, S., Schmeier, S., Arens, R.G.: Message classification in the call center. In: Proc. Sixth Conference on Applied Natural Language Processing, pp. 158–165. ACL (2000)
3. Lapalme, G., Kosseim, L.: Mercure: Towards an automatic e-mail follow-up system. IEEE Computational Intelligence Bulletin 2(1), 14–18 (2003)
4. Scheffer, T.: Email answering assistance by semi-supervised text classification. In: Intelligent Data Analysis, vol. 8(5), pp. 481–493. IOS Press, Amsterdam (2004)
5. Malik, R., Subramaniam, V., Kaushik, S.: Automatically Selecting Answer Templates to Respond to Customer Emails. In: Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, pp. 1659–1664 (2007)
6. Weng, S.S., Liu, C.K.: Using text classification and multiple concepts to answer e-mails. Expert Systems with Applications 26(4), 529–543 (2004)
7. Marom, Y., Zukerman, I.: Towards a Framework for Collating Help-desk Responses from Multiple Documents. In: Proceedings of the IJCAI Workshop on Knowledge and Reasoning for Answering Questions, Edinburgh, Scotland, pp. 32–39 (2005)
8. Kosseim, L., Beauregard, S., Lapalme, G.: Using information extraction and natural language generation to answer e-mail. Data & Knowledge Engineering 38, 85–100 (2001)
9. Sneiders, E.: Automated FAQ Answering with Question-Specific Knowledge Representation for Web Self-Service. In: Bello, L.L., Iannizzotto, G. (eds.) Proc. 2nd International Conference on Human System Interaction, pp. 298–305. IEEE, Los Alamitos (2009)
10. Tang, J., Li, H., Cao, Y., Tang, Z.: Email Data Cleaning. In: Proc. Eleventh ACM SIGKDD Internationl Conference on Knowledge Discovery in Data Mining, Chicago, Illinois, USA, pp. 489–498. ACM, New York (2005)
11. Dalianis, H.: Evaluating a Spelling Support in a Search Engine. In: Andersson, B., Bergholtz, M., Johannesson, P. (eds.) NLDB 2002. LNCS, vol. 2553, pp. 183–190. Springer, Heidelberg (2002)
12. Golding, A.R., Roth, D.: A Winnow-Based Approach to Context-Sensitive Spelling Correction. In: Machine Learning, vol. 34, pp. 107–130. Springer, Netherlands (1999)
13. Mays, E., Damerau, F.J., Mercer, R.L.: Context Based Spelling Correction. Information Processing & Management 27(5), 517–522 (1991)
14. Brill, E., Moore, R.C.: An Improved Error Model for Noisy Channel Spelling Correction. In: Proc. 38th Annual Meeting on Association for Computational Linguistics, Hong Kong, pp. 286–293. ACL (2000)

A System to Control Language for Oral Communication

Laurent Spaggiari[1] and Sylviane Cardey[2]

[1] Human Factors Dept., EDYDNX, section 527, M0151/0, Airbus Operations SAS, 316 route de Bayonne, F-31060 Toulouse, France
[2] Centre Tesnière, Université de Franche-Comté, UFR SLHS, 30 rue Mégevand, F-25030 Besançon Cedex, France
laurent.SPAGGIARI@airbus.com, sylviane.cardey@univ-fcomte.fr

Abstract. In this paper we discuss the use of controlled languages not for written texts but for oral communication which has never been done before, and this in safety critical domains. Interference between languages could effectively cause accidents due to misunderstanding of messages whatever they are. We discuss how firstly we could automatically detect eventual possibilities of misunderstanding due to mispronunciation or bad interpretation and secondly how to prevent these problems by using controlled languages. We show that our methodology, which is intensional in nature, is much more productive than working in extension.

Keywords: Controlled language, oral communication, language interferences.

1 Introduction

In this paper we discuss the use of controlled languages not for written texts but for oral communication which has never been done before, and this in safety critical domains. Interference between languages could effectively cause accidents due to misunderstanding of messages whatever they are. Though research concerning interferences between different languages has been carried out at Airbus Operations SAS and in Centre Tesnière, this has been written-text oriented and not oral-text oriented. This paper is concerned with Airbus Operations SAS and Centre Tesnières' research into helping in the design of controlled languages which are optimized for computer-human communication.

We discuss how firstly we could automatically detect eventual possibilities of misunderstanding due to mispronunciation or bad interpretation and secondly how to prevent these problems by using controlled languages. We show that our methodology, which is intensional in nature, is much more productive than working in extension.

2 Natural Language

Whether it be oral, written or signed, a communication will be considered as successful and efficient when the received message complies with the mental process used for reconstructing and interpreting the information within the message. However, natural language not only allows everyone to create many variations for the same

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 393–400, 2010.

expression – *Paul sold the car* versus *The car was sold by Paul* – but it is also intrinsically ambiguous from the following points of view:

- semantics (word): *Parle-moi de ta nouvelle pièce*
  (Tell me about your new room/coin/play)
- semantics (sentence): *Paul donne un os à son chien pour s'amuser*
  (Paul gives his dog a bone for his own fun/for the dog to play with)
- phonetics (word): *il est assis sur [letal*O~*]*
  (he is riding (sitting on) the stallion/he is squatting on his heels)
- phonetics (sentence): *[sEtwazola]*
  (you are Zola/that guy/that bird)
- syntax (word): *Paul préfère les gâteaux au chocolat*
  (Paul prefers chocolate cakes/Paul prefers cakes to chocolate)
- syntax (sentence): *Flying planes can be dangerous*
  (this can be a dangerous work/ aircraft above your head can be dangerous)
- pragmatics (word): *Le magasin est ouvert le dimanche*
  (the store is open only/even on Sunday)
- pragmatics (sentence): *Paul a dit qu'il viendrait*
  (Paul said he (Paul/Jean) would come)

In certain highly technical domains such as nuclear/surgery/chemistry/aeronautics, etc. safety is crucial and some situations require from the operators immediate corrective actions and to succeed in their tasks, they needs to understand fully and immediately the situation, i.e. what they are expected to do, the consequences, etc. In this respect, natural language, too broad, too variable cannot be used.

3 Controlled Languages

Contrary to natural language, controlled languages (CLs) are favoured by industry because they refer to systems that limit the number of core vocabulary words, of applicable grammar and stylistic rules. "Industry does not need Shakespeare or Chaucer; industry needs clear, concise communicative writing – in one word Controlled Language" [1]. Their objective is to reduce ambiguities, complexity, colloquialisms and synonyms in order to improve consistency, readability, translatability and retrieval of information [2]. "Consistency is one of the most basic usability principles [...] the same information [...] should be formatted in the same way to facilitate recognition" [3].

This concept of CL is not really new. Indeed, in the 1930s, C. K. Ogden, a linguist, developed British American Scientific International Commercial English to help students write in a clearer way and to make the non Anglophone students' training easier [4]. Since then, many of CLs have been developed for different purposes, e.g. TAUM meteo, Air/Sea/police speak, Douglas Aircraft, ScaniaSwedish, Caterpillar Fundamental/Technical English, Kodak International Service Language, and, the most widely used for writing aircraft maintenance procedures, ASD-STE 100 (formerly known as AECMA Simplified English [5]); for a more complete overview, refer to [6]. These CLs are not "simple" or "baby" English but true simplified English.

Although English is a very productive natural language for the creation of CLs as it is the current international language used for trade and science, other languages such as German, Chinese, Swedish and French are also used for the creation of CLs.

4 The Oral Aspect

Creators of CLs usually base their grammar restrictions on well-established writing principles (e.g. "write short sentences with only one topic; avoid passive form ..."). Furthermore, despite the fact that these languages do not have many rules in common [7], they do share one main characteristic: they deal with the written aspects of language, and not with the oral.

It is exactly this oral aspect that we address here. The fact is that messages are not only read but can also be heard using synthetic/recorded voices in nuclear plants and airports for example. Because the receiver of the message may not have the same mother tongue as the one he/she hears and because one cannot expect him/her to master it, a syntactically and lexically controlled message may not be sufficient. Indeed, when looking at the following pairs, one can easily imagine the potential consequences in case of a misunderstanding:

- *increase the temperature* versus *decrease the temperature*
- *the gear is uplocked* versus *the gear is unlocked*
- In French for an Anglophone *"dessus" and "dessous"* will sound the same

As a further example, ambiguities can result from English phonemes not present in Thai, as is illustrated in Table 1 [8]. We do not enter into the complexity of the matter here, but we can already see that *half* for example becomes *harp* as well as *ball,* which is pronounced *born*.

Table 1. Ambiguities resulting from English phonemes not present in Thai (the phonetic transcriptions are in SAMPA)

English phonemes	Sound in Thai	Ambiguities
[T]	[t], [d]	birth → bird
[D]	[d]	they → day
[f]	[p]	half → harp
[v]	[w]	vine → wine
[s]	[d]	bus → bud
[z]	[t]	buzz → but
[l]	[n]	ball → born
[r]	[l]	free → flee

5 A System to Help When Creating Sentences That Are to Be Pronounced

Based on the observations in the previous section, we have devised a system that can help when creating sentences that are to be pronounced. This system has the ability to detect within a list not only all the homophones (e.g. *night knight*) and the minimal pairs (e.g. *brake brain*) but also quasi homophones (e.g. *increase decrease*) for a proposed word, according to the source language (North American English).

5.1 The Database

The database we used for checking the pronunciation is the Carnegie Mellon University Pronouncing Dictionary, also known as 'cmudict'. This dictionary is a public domain machine-readable pronunciation dictionary for North American English that contains over 130,000 words and their phonetic transcriptions. The pronunciation of the words is encoded using a modified form of the Arpabet system, each phoneme having a unique code (e.g. *ABRACADABRA AE2 B R AH0 K AH0 D AE1 B R AH0).* This dictionary is used in different projects such as the Festival speech synthesis system and also the CMU Sphinx speech recognition system.

As a result of this database, our system was able to retrieve all the words with the same pronunciation. However, we also wanted to obtain quasi homophones (e.g. *increase decrease*) from this list. So we devised an algorithm that looks at the phonetic differences between words.

5.2 The Algorithm

The algorithm performs the following steps:

- Calculation of the number of phonemes for the submitted word (11 for *abracadabra*)
- Retrieval from the database of all the words that have:
    - The same number of phonemes
    - The same number +1 of phonemes
    - The same number -1 of phonemes
- Calculation of the similarity (number of different phonemes in the same order) between the submitted word and the retrieved words
- Calculates the proximity between the two phonetic strings using the Levenshtein distance function.

Levenshtein distance is defined as the minimum number of necessary characters to be changed, inserted, or modified for transforming a string into another one. It is commonly used for spelling checking, speech recognition, DNA analysis, etc.

The system works by requesting a check for a specific word, one by one. So, it is not possible to get statistics on the numbers of pairs of words retrieved for a specific language. The algorithm is able to retrieve any string of any length (monosyllabic to very long words). The number of retrievals decreases with the length of the submitted word. Also, the words retrieved are sometimes irrelevant (different enough to be not mistaken). This is due to the fact that we weakened the constraints. Indeed, a difference of 2 phonemes is considered in our algorithm. Also, we did consider the phoneme itself as a whole and not as a sum of phonological features. We think the retrievals will be more relevant when we will look at the divergences of phonological features and only consider one phoneme of difference instead of 2 as at present.

5.3 Oral Communication Involving Different Mother Tongues

The receiver of the message can have a different mother tongue from English. Consequently, some phonemes may not exist for him or her. In this case, what will the receiver understand? Assuming the fact that he/she will reconstruct words using existing phonemes in their own language, the system should replace the phonemes by

those existing in another language and check in the database if homophones or quasi homophones exist.

To do this, we devised a resource that gives, for each language (Arabic, French and Chinese for the moment), the list of non-existing English phonemes and their counterpart(s). Table 2 illustrates an extract from this table.

Table 2. Non-existing English phonemes and their counterpart(s) (extract)

From English to		
Phonemes	**Replaced by**	**Language**
Ax	Aa	Arabic
Ax	Ae	Arabic
Ax	Ao	Arabic

As a result of this resource, our system is now able, depending on the language selected, to reconstruct the pronunciation and then retrieve all the words with the same pronunciation in English. An illustration of the system's interface showing the results for tomato is shown in Fig. 1.

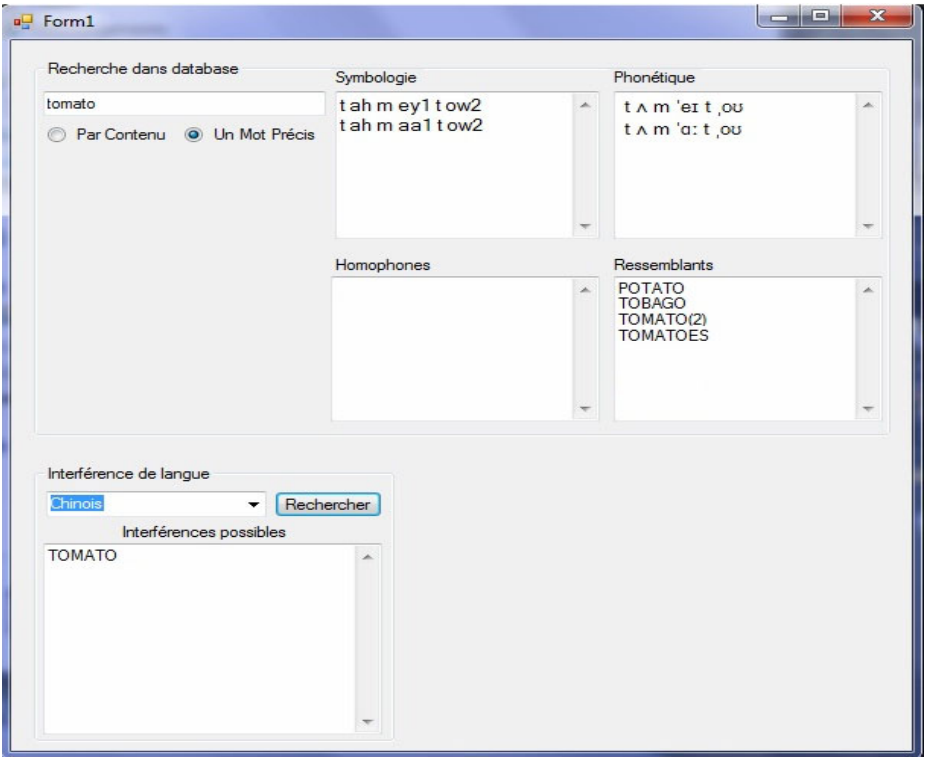


Fig. 1. Screen shot of the 'System to Help when Creating Sentences that are to be Pronounced' for the word *tomato*

6 Results and Improvements

The system behaves as intended because:

- it retrieves all the homophones for a term (*feel* (*F IY1 L*) → *fiel, feil, foell*).
- it retrieves all the minimal pairs for a term (*feel* (*F IY1 L*) → *fail, fall, feat, feed, fees, fell, file, fill, foal, foil, fool, foul, fowl, full, peal, peel*).
- it retrieves quasi homophones for a term depending on the language selected (*thought* (*TH AO1 T*) → for Chinese: *fought, sawed, sod, sought* and for French: *fought, sought, taught, taut, tot*)

Using this information, one can easily decide, when creating a spoken message, if one can use a word or if one should change it (e.g. use *reduce* instead of *decrease*) or even reformulate the whole sentence.

However, when looking just at English, some of the words that were retrieved could be avoided as they are different enough not to be mistaken. We think that the reasons for this lie in the constraints (insufficient) we have applied for calculating the similarity. Indeed, we performed this calculation by counting the number of different phonemes between two words. Also, to be able to retrieve for example *increase decrease*, we had to consider an acceptable number of 2 differences whatever they are. As a consequence, many words with 2 differences are retrieved. A better way would be to take into account for each phoneme its phonological features, and to count the different ones to get a much more precise result. For example, we would continue to take a difference of 2 phonemes as a maximum, but, reducing this time the maximum number of differences allowed between features, we would reduce the numbers of results. The 2 different phonemes in *increase decrease* share the same phonological features except one: nasal vs. oral.

Another improvement would consist in considering the whole sentence, that is to say, to consider the assimilations that occur between the words once these are put together.

Finally, the phonetics of the reconstructed sentences could be saved in an ssml file [9]:

```
<speak version="1.0"
xmlns="http://www.w3.org/2001/10/synthesis"
xml:lang="en-US">
<phoneme ph="tʌmˈeɪɾˌoʊ">tomato</phoneme>
</speak>
```

This file can easily be enriched with plenty of information concerning prosody and style (voice, emphasis, break, pitch, speaking rate and volume of the speech output), text structure etc. This would allow us to use this file as an entry to obtain these phonetic strings pronounced by a synthetic voice (for example Microsoft US English Anna). An illustration of the system's interface enriched for voice synthesis (word by word only) is shown in Fig. 2.

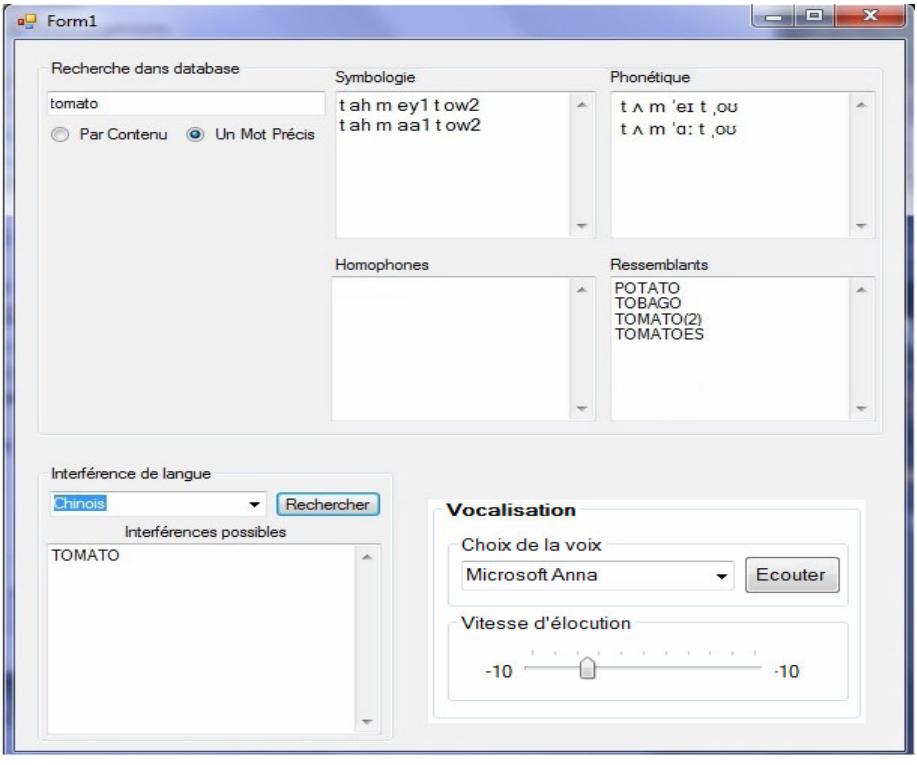


Fig. 2. Screen shot of the 'System to Help when Creating Sentences that are to be Pronounced' enriched for voice synthesis

7 Conclusion

We have seen in this paper why language interferences have to be avoided and we have proposed a methodology and a system to find and solve these problems. Some work about interference had already been done in our laboratory with native speakers but our system automatically detecting possible interference revealed itself much more efficient. Much work still has to be done at the level of the boundaries between words but the methodology which consists in working at the level of phonemes and distinctive features rather than trying to find individual words seems to be more productive and easier to generalise for solving the problems of interferences which are due to bad pronunciation or bad interpretation. The methodology used allows tracing back to the cause of the problems which is essential in safety critical applications. The results of this research can be applied to different domains. This is because the specific data (i.e. lexicon by domain) is tested against the general pronunciation dictionary as we cannot know the level of English people have. So the methodology is not domain dependant and can be applied to any specific domain as long as the domain has its own dictionary.

References

1. Goyvaerts, P.: Controlled English, Curse or Blessing? A User's Perspective. In: Proceedings of CLAW 1996, Leuven, Belgium, March 26-27, pp. 137–142 (1996)
2. Huijsen, W.O.: Controlled Language: An Introduction. In: Proceedings of CLAW 1998, Pittsburg, USA, May 21-22, pp. 1–15 (1998)
3. Nielsen, J.: Usability Engineering. Academic Press Ltd., London (1993)
4. Ogden, C.K.: Basic English: A General Introduction with Rules and Grammar, 8th edn. (1940); PSYCHE 29, K. Paul, London, England, (1930)
5. AECMA Simplified English Association Européenne des Constructeurs de Matériel Aérospatial. A Guide for The Preparation of Aircraft Maintenance Documentation. The International Aerospace Maintenance Language', Gulledelle, Brussels, Belgium (1995)
6. Gavieiro-Villatte, E., et al.: Open-Ended Overview of Controlled Languages. In: BULAG 24, Besançon, France (1999)
7. O'Brien, S.: Controlling Controlled English: An Analysis of Several Controlled Language Rule Sets. In: Proceedings of EAMT-CLAW 2003, Dublin, Ireland, May 15-17, pp. 105–114 (2003)
8. Cardey, S.: How to avoid interferences with other languages when constructing a spoken controlled language. In: Proceedings of the International Conference « La comunicazione parlata/Spoken Communication », Naples, Italy, February 23 -25 (2006)
9. Speech Synthesis Markup Language (SSML) Version 1.1, http://www.w3.org/TR/2010/PR-speech-synthesis11-20100223/

Robust Semi-supervised and Ensemble-Based Methods in Word Sense Disambiguation

Anders Søgaard and Anders Johannsen

Centre for Language Technology
University of Copenhagen
Njalsgade 140–142
DK-2300 Copenhagen S
{soegaard,ajohannsen}@hum.ku.dk

Abstract. Mihalcea [1] discusses self-training and co-training in the context of word sense disambiguation and shows that parameter optimization on individual words was important to obtain good results. Using smoothed co-training of a naive Bayes classifier she obtains a 9.8% error reduction on Senseval-2 data with a fixed parameter setting. In this paper we test a semi-supervised learning algorithm with no parameters, namely tri-training [2]. We also test the random subspace method [3] for building committees out of stable learners. Both techniques lead to significant error reductions with different learning algorithms, but improvements do not accumulate. Our best error reduction is 7.4%, and our best absolute average over Senseval-2 data, though not directly comparable, is 12% higher than the results reported in Mihalcea [1].

Keywords: co-training, tri-training, word sense disambiguation.

1 Introduction

Word sense disambiguation (WSD) is the task of deciding which sense a word has in a particular context. The Senseval-2 shared task provides labeled data that can be used for supervised learning of WSD of 29 English nouns. This data set was used by Mihalcea [1] in a line of experiments using inference from unlabeled data in addition to the Senseval-2 data, namely instances drawn from the British National Corpus.

The baseline in Mihalcea [1] is a naive Bayes classifier trained on the labeled data. She then considers the potential of self-training and co-training algorithms for making use of unlabeled data [4]. She first shows that performance is very sensitive to parameter setting, but nevertheless, co-training leads to a significant error rate reduction of 9.8% with a global parameter setting, which specifies the number of iterations, growth size, and pool size. In particular, co-training is run twice with a pool of 5000 data points, selecting the 50 points most confidently labeled.

In this work we consider a parameter-free semi-supervised learning algorithm introduced in Li and Zhou [2]. The algorithm is described in Sect. 2. Sect. 3

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 401–405, 2010.

introduces a method for constructing ensembles of classifiers that form robust and accurate end classifiers, namely random subspaces [3]. In Sect. 4 we apply tri-training and random subspaces to supervised classifiers trained on Senseval-2 data, incl. naive Bayes, decision stumps, PART and logistic boosting. In Sect. 5 we discuss our results and conclude.

The reported preprocessing of the data in Mihalcea [1] is very complicated, and the preprocessed data is no longer available (Mihalcea, p.c.). Consequently, results reported here are not directly comparable. Moreover, some test examples in the Senseval-2 have multiple labels, and to simplify things we only evaluate our algorithms on test examples with a unique label.

2 Tri-training

This section presents the tri-training algorithm originally proposed by Li and Zhou [2].

Let L denote the labeled data and U the unlabeled data. Assume that three classifiers c_1, c_2, c_3 (same learning algorithm) have been trained on three bootstrap samples of L. In tri-training, an unlabeled datapoint in U is now labeled for a classifier, say c_1, if the other two classifiers agree on its label, i.e. c_2 and c_3. Two classifiers inform the third. If the two classifiers agree on a labeling, there is a good chance that they're right. The algorithm stops when the classifiers no longer change. The three classifiers are combined by majority voting. Li and Zhou [2] show that under certain conditions the increase in classification noise rate is compensated by the amount of newly labeled data points.

The most important condition is that the three classifiers are diverse. If the three classifiers are identical, tri-training degenerates to self-training. Diversity is obtained in Li and Zhou [2] by training classifiers on bootstrap samples. In their

1: **for** $i \in \{1..3\}$ **do**
2: $S_i \leftarrow bootstrap\_sample(L)$
3: $c_i \leftarrow train\_classifier(S_i)$
4: **end for**
5: **repeat**
6: **for** $i \in \{1..3\}$ **do**
7: **for** $x \in U$ **do**
8: $L_i \leftarrow \emptyset$
9: **if** $c_j(x) = c_k(x)(j, k \neq i)$ **then**
10: $L_i \leftarrow L_i \cup \{(x, c_j(x)\}$
11: **end if**
12: **end for**
13: $c_i \leftarrow train\_classifier(L \cup L_i)$
14: **end for**
15: **until** none of c_i changes
16: apply majority vote over c_i

Fig. 1. Tri-training (Li and Zhou, 2005)

experiments, they consider classifiers based on the C4.5 algorithm, BP neural networks and naive Bayes classifiers. The algorithm is sketched in a simplified form in Figure 1; see Li and Zhou [2] for all the details.

Tri-training has to the best of our knowledge not been applied to WSD before, but it has been applied to other NLP classification tasks, incl. Chinese chunking [5] and question classification [6].

3 Random Subspaces

Random subspaces was introduced in Ho [3]. The idea is simple: Randomly select a subset of the components of the feature vector and train a classifier. In other words, from a d-dimensional data set we project n new k-dimensional data sets by random projection. Each new data set is given as input to the learning algorithm, and the base classifiers are combined to form a stronger end classifier by majority voting. One weakness with this method is that some of the subspaces may lack the ability to separate different classes. For this reason we expect random subspaces to perform well with boosting algorithms; see García-Pedrajas and Ortiz-Boyer [7] for a similar point.

4 Experiments

4.1 Data

The data sets were prepared to be as close as possible to the data sets used in Mihalcea [1]. Briefly put, we use a small set of local features, incl. context word forms and POS tags, and a larger set of global features, i.e. a small bag of (content) words. Unlike Mihalcea [1], we do not use collocations as global features, and we only use a single pool of unlabeled data points rather than pruning them by collocations. See Mihalcea [1] for more details. The complete list of words used in Mihalcea [1], i.e. the nouns in the Senseval-2 data:

Table 1. English nouns in Senseval-2

art	authority	bar	bum	chair	channel	child	church
circuit	day	detention	dyke	facility	fatigue	feeling	grip
hearth	holiday	lady	material	mouth	nation	nature	post
restraint	sense	space	stress	yew			

The average number of labeled examples for each word is about 100 with an additional pool for testing of about half that figure. The average number of unlabeled examples in our experiments is 14,355, compared to 7,085 in Mihalcea [1]. Mihalcea [1] prunes the unlabeled data by collocations to avoid noise, but using the raw unlabeled data makes it easier to reproduce our results.

4.2 Learning Algorithms

We consider three baseline learning algorithms in our experiments. Sect. 3 motivated our choice of using logistic boosting [8]. Briefly put, logistic boosting considers the more well-known boosting algorithms as generalized additive models and applies the cost functional of logistic regression. Naive Bayes is the standard choice in WSD and is known to perform reasonably well on Senseval-2 data. We also ran our algorithms on decision stumps because they are fundamental to a wide range of learning algorithms, incl. logistic boosting. Finally, we include a rule-based learning algorithm, PART [9], because of their intuitive nature and potential usefulness in more descriptive computational linguistics.

4.3 Results

Our results using the tri-training algorithm on Senseval-2 data are listed below. We use self-training until convergence as our semi-supervised baseline; see e.g. Abney [4]. We only present average accuracy rather than accuracies on the 29 individual words. Δ is the absolute difference between our baseline and tri-training (resp. random subspaces):

Table 2. Results for tri-training

learner	baseline	self-training	tri-training	Δ
LogitBoost	65.56	66.39	**66.43**	0.87
naive Bayes	**64.33**	64.09	62.74	-1.59
PART	60.37	60.57	**60.84**	0.47
DecisionStump	58.23	58.59	58.86	0.63

Tri-training leads to reasonable error reductions when applied to logistic boosting (2.5%) and PART (1.2%), in light of our relatively strong baselines, but not when applied to naive Bayes. Our results using random subspaces on Senseval-2 data are listed here:

Table 3. Results for random subspaces

learner	baseline	bagging	random subspaces	Δ
LogitBoost	65.56	66.80	**68.11**	2.55
naive Bayes	**64.33**	63.61	64.01	-.32
PART	60.37	62.68	**63.16**	2.79
DecisionStump	58.23	56.62	58.65	0.42

The result obtained by random subspaces over logistic boosting even takes us considerably beyond what can be obtained with linear support vector machines. We use bagging as our ensemble-based baseline.

Combining Tri-Training and Random Subspaces? Somewhat surprisingly performance degrades if we try to combine tri-training and random subspaces. We tri-trained a random subspace model with logistic boosting, but accuracy dropped by more than 2 percentage points (to 66.00%) compared to our random subspaces baseline.

5 Conclusion

Tri-training and random subspaces are, by and large, robust methods for boosting supervised classifiers in the context of word sense disambiguation. Naive Bayes is probably too stable to be amenable to tri-training, although Li and Zhou [2] report positive results, and it is clear why random subspaces hurts the performance of naive Bayes: Since naive Bayes implements a strong independence assumption between features and consult each feature independently, there is little to gain from randomly constructed subspaces. Both tri-training and random subspaces work particularly well with logistic boosting. It seems logistic boosting helps overcome the potential weaknesses of random subspaces, and it is, unlike support vector machines, for example, sensitive enough to bootstrap samples to be amenable to tri-training.

Generally, we reported competitive results on the Senseval-2 (68.11%) and obtained a 7.4% error reduction wrt. logistic boosting. Since the data set is not directly comparable to the data set used in Mihalcea [1] we plan to reimplement smoothed co-training for direct comparison. We also plan to compare tri-training to semi-supervised support vector machines [10].

References

1. Mihalcea, R.: Co-training and self-training for word sense disambiguation. In: CONLL, Boston, MA (2004)
2. Li, M., Zhou, Z.H.: Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering 17(11), 1529–1541 (2005)
3. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
4. Abney, S.: Semi-supervised learning for computational linguistics. Chapman and Hall, Boca Raton (2008)
5. Chen, W., Zhang, Y., Isahara, H.: Chinese chunking with tri-training learning. In: Matsumoto, Y., Sproat, R.W., Wong, K.-F., Zhang, M. (eds.) ICCPOL 2006. LNCS (LNAI), vol. 4285, pp. 466–473. Springer, Heidelberg (2006)
6. Nguyen, T., Nguyen, L., Shimazu, A.: Using semi-supervised learning for question classification. Journal of Natural Language Processing 15, 3–21 (2008)
7. García-Pedrajas, N., Ortiz-Boyer, D.: Boosting random subspace method. Neural Networks 21(9), 1344–1362 (2008)
8. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2), 337–407 (2000)
9. Frank, E., Witten, I.: Generating accurate rule sets without global optimization. In: The 15th International Conference on Machine Learning (1995)
10. Sindhwani, V., Keerthi, S.: Large scale semi-supervised linear SVMs. In: ACM SIGIR, Seattle, WA (2006)

The Effect of Semi-supervised Learning on Parsing Long Distance Dependencies in German and Swedish

Anders Søgaard and Christian Rishøj

Center for Language Technology
University of Copenhagen
Njalsgade 140–142
DK-2300 Copenhagen S
{soegaard,crjensen}@hum.ku.dk

Abstract. This paper shows how the best data-driven dependency parsers available today [1] can be improved by learning from unlabeled data. We focus on German and Swedish and show that labeled attachment scores improve by 1.5%-2.5%. Error analysis shows that improvements are primarily due to better recovery of long distance dependencies.

Keywords: dependency parsing, semi-supervised learning, long distance dependencies.

1 Introduction

Rimell et al. [2] argue that long distance dependencies are particularly interesting in parser evaluation, since they provide a strong test of the parser's knowledge of grammar, and since recovering long distance dependencies is necessary to completely represent the underlying predicate-argument structure of the sentence, useful for applications such as question answering and information extraction. Rimell and Clark show that state-of-the-art constituent parsers have accuracies below 50% on a new dataset of English unbounded dependencies.

The purpose of this paper is two-fold: (i) It is shown that it is possible to improve the accuracy of the best available dependency parsers for German and Swedish by learning from unlabeled data. (ii) It is shown that improvements are primarily due to better recovery of long distance dependencies. In particular it is shown that a novel semi-supervised learning algorithm called generalized tri-training is able to improve labeled attachment scores (LASs) on standard datasets by 1.72% (German) and 2.36% (Swedish) in general. If we limit attention to long distance dependencies (≥ 7), however, increases in F-score are even more dramatic, i.e. 5.09% (German) and 8.58% (Swedish).

Semi-supervised learning of structured variables is a difficult problem that has received considerable attention recently, but most results have been negative [3]. This paper uses stacked learning [4] to reduce structured variables, i.e. dependency graphs, to multinomial variables, i.e. attachment and labeling decisions, which are easier to manage in semi-supervised learning scenarios.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 406–417, 2010.

Ensemble-based methods such as stacked learning are used to reduce the instability of classifiers, to average out their errors and to combine the strengths of diverse learning algorithms. Ensemble-based methods have attracted a lot of attention in dependency parsing recently [5,6,7,1,8,9]. Nivre and McDonald [7] were first to introduce stacking in the context of dependency parsing.

This paper applies a generalization of tri-training [10], a form of co-training that trains an ensemble of three learners on labeled data and runs them on unlabeled data, to two classification problems, attachment and labeling, that together approximate dependency parsing. Semi-supervised dependency parsing has attracted a lot of attention recently [11,12,13], but there has, to the best of our knowledge, been no previous attempts to apply tri-training or related combinations of ensemble-based and semi-supervised methods to any of these tasks, except for the work of Sagae and Tsujii [14]. However, tri-training has been applied to Chinese chunking [15] and question classification [16].

We compare generalized tri-training to the original tri-training algorithm and to semi-supervised support vector machines [17].

Sect. 2 first introduces the dependency parsing problem and defines stacked learning. Stacked learning is then generalized to dependency parsing, and we describe how stacked dependency parsers can be further stacked as input for two end classifiers that can be combined to produce dependency structures. These two classifiers will learn multinomial variables (attachment and labeling) from a combination of labeled data and unlabeled data using a generalization of the tri-training algorithm presented in Li and Zhou [10]. Sect. 2 also introduces generalized tri-training.

Sect. 3 describes our experiments. We describe the data sets, and how the unlabeled data were prepared. Sect. 4 presents our results. Sect. 5 presents an error analysis and shows that improvements are primarily due to better recovery of long distance dependencies, and Sect. 6 concludes the paper.

2 Background

2.1 Dependency Parsing

Dependency parsing models a sentence as a tree where words are vertices and grammatical functions are directed edges (dependencies). Each word thus has a single incoming edge, except one called the root of the tree. Dependency parsing is thus a structured prediction problem with trees as structured variables. Each sentence has exponentially many possible dependency trees. Our observed variables are sentences with words labeled with part-of-speech tags. The task for each sentence is to find the dependency tree that maximizes an objective function which in our case is learned from a combination of labeled and unlabeled data.

More formally, a dependency tree for a sentence $x = w_1, \ldots, w_n$ is a tree $T = \langle \{0, 1, \ldots, n\}, A \rangle$ with $A \subseteq V \times V$ the set of dependency arcs. Each vertex corresponds to a word in the sentence, except 0 which is the root vertex, i.e. for any $i \leq n$ $\langle i, 0 \rangle \notin A$. Since a dependency tree is a tree it is acyclic. A tree

is projective if every vertex has a continuous projection, i.e. if and only if for every arc $\langle i, j\rangle \in A$ and node $k \in V$, if $i < k < j$ or $j < k < i$ then there is a subset of arcs $\{\langle i, i_1\rangle, \langle i_1, i_2\rangle, \ldots, \langle i_{k-1}, i_k\rangle\} \in A$ such that $i_k = k$. The German and Swedish data sets used in our experiments below have 27.8%, resp. 9.8%, non-projective dependency trees.

2.2 Stacked Dependency Parsing

Stacked generalization, or simply *stacking*, was first proposed by Wolpert [4]. Stacking is an ensemble-based learning method where multiple weak classifiers are combined in a strong end classifier. The idea is to train the end classifier directly on the predictions of the input classifiers.

Say each input classifier c_i with $1 \leq i \leq n$ receives an input $\mathbf{x}$ and outputs a prediction $c_i(\mathbf{x})$. The end classifier then takes as input $\langle \mathbf{x}, c_1(\mathbf{x}), \ldots, c_n(\mathbf{x})\rangle$ and outputs a final prediction $c_0(\langle \mathbf{x}, c_1(\mathbf{x}), \ldots, c_n(\mathbf{x})\rangle)$. Training is done by cross-validation. In sum, stacking is training a classifier on the output of classifiers.

Stacked learning can be generalized to structured prediction tasks such as dependency parsing. Architectures for stacking dependency parsers typically only use one input parser, but otherwise the intuition is the same: the input parser is used to augment the dependency structures that the end parser is trained and evaluated on.

Nivre and McDonald [7] first showed how the MSTParser [18] and the MaltParser [19] could be improved by stacking each parser on the predictions of the other. Martins et al. [1] generalized their work, considering more combinations of parsers, and stacking the end parsers on non-local features from the predictions of the input parser, e.g. siblings and grand-parents. In this work we parse the German and Swedish data sets from the CONLL-X Shared Task and use three stacked dependency parsers for each language:

	parser1 (p_1)	parser2 (p_2)	parser3 (p_3)
Ge	mst2	malt/mst2(D)	malt/mst1(E)
Sw	mst2/mst2(D)	malt/mst2(D)	malt/mst1(A)

The notation "malt/mst2" means that the second-order MSTParser has been stacked on MaltParser. The capital letters refer to feature configurations. Configuration A only stacks the level 1 parser on the predicted edges of the level 0 parser (along with the input features). Configuration D stacks a level 1 parser on several (non-local) features of the predictions of the level 0 parser: the predicted edge, siblings, grand parents and predicted head of candidate modifier if predicted edge is 0. Configuration E stacks a level 1 parser on the features in configuration D and all the predicted children of the candidate head. The chosen parser configurations are those that performed best in Martins et al. [1].

There are two reasons that our input parsers perform slightly worse than those reported on in Martins et al. [1]: (i) We use about 5,000 tokens of the training data for development. (ii) For both datasets, we used projective rather than pseudoprojective parsing in MaltParser. For MSTParser (level 0), we also

reduced training time by iterating three times over the German data rather than 10 times as in Martins et al. [1].

2.3 Stacking Stacked Dependency Parsing

The input features of the input classifiers in stacked learning $\mathbf{x}$ can of course be removed from the input of the end classifier. It is also possible to stack stacked classifiers. This leaves us with four strategies for recursive stacking; namely to constantly augment the feature set, with level n classifiers trained on the predictions of the classifiers at all $n-1$ lower levels with or without the input features $\mathbf{x}$, or simply to train a level n classifier on the predictions of the level $n-1$ classifiers with or without $\mathbf{x}$.

In this work we stack stacked dependency parsers by training classifiers on the output of three stacked dependency parsers and POS tags. Consequently, we use one of the features from $\mathbf{x}$, since this led to better results on development data. Note that we train classifiers and not parsers on this new level 2.

The reduction is done the following way: First we train a classifier on the relative distance from a word to its head to induce attachments. For example, we may obtain the following features from the predictions of our level 1 parsers:

label	p_1	p_2	p_3	POS
1	1	-1	1	NNP
0	0	0	0	VBD

In the second row all input parsers, p_{1-3} in column 2–4, agree that the verb (VBD) is the root of the sentence. Column 1 tells us that this is correct. In the first row, two out of three parsers agree on attaching the noun (NNP) to the verb, which again is correct. We train level 2 classifiers on feature vectors produced this way. Note that oracle performance of the ensemble is no upper bound on the accuracy of a classifier trained on level 1 predictions this way, since a classifier may learn the right decision from three wrong predictions and a POS tag.

Second we train a classifier to predict dependency relations. Our feature vectors are similar to the ones just described, but now contain dependency label predictions, e.g.:

label	p_1	p_2	p_3	POS
SBJ	SBJ	SBJ	SBJ	NN
ROOT	ROOT	ROOT	COORD	VBN

2.4 Generalized Tri-training

Tri-training was originally introduced in Li and Zhou [10]. The method involves three learners that inform each other.

Let L denote the labeled data and U the unlabeled data. Assume that three classifiers c_1, c_2, c_3 have been trained on L. In the original algorithm, the three

classifiers are obtained by applying the same learning algorithm to three bootstrap samples of the labeled data; but in generalized algorithms, three different learning algorithms are used. An unlabeled datapoint in U is labeled for a classifier, say c_1, if the other two classifiers agree on its label, i.e. c_2 and c_3. Two classifiers inform the third. If the two classifiers agree on a labeling, we assume there is a good chance that they are right. In the original algorithm, learning stops when the classifiers no longer change; in generalized tri-training, a fixed stopping criterion estimated on development data is used. The three classifiers are combined by majority voting. Li and Zhou [10] show that under certain conditions the increase in classification noise rate is compensated by the amount of newly labeled data points.

The most important condition is that the three classifiers are diverse. If the three classifiers are identical, tri-training degenerates to *self-training*. As already mentioned, Li and Zhou [10] obtain this diversity by training classifiers on bootstrap samples. In their experiments, they consider classifiers based on decision trees, BP neural networks and naïve Bayes inference.

In this paper we generalize the tri-training algorithm and use three different learning algorithms rather than bootstrap samples to create diversity: a naïve Bayes algorithm (no smoothing), random forests [20] (with 100 unpruned decision trees) and an algorithm that induces unpruned decision trees. The three algorithms are fast, represent different degrees of stability, and no parameter tuning is necessary. The overall algorithm is sketched in Figure 1.

```
1: for i ∈ {1..3} do
2:    c_i ← train_classifier(l_i, L)
3: end for
4: repeat
5:    for i ∈ {1..3} do
6:       for x ∈ U do
7:          L_i ← ∅
8:          if c_j(x) = c_k(x)(j, k ≠ i) then
9:             L_i ← L_i ∪ {(x, c_j(x)}
10:         end if
11:      end for
12:      c_i ← train_classifier(L ∪ L_i)
13:   end for
14: until stopping criterion is met
15: apply c_i
```

Fig. 1. Generalized tri-training

Our predictions are those of the random forests classifier after a fixed number of rounds optimized on development data. On development data this led to slightly better results than majority votes.

3 Experiments

3.1 Data

We use the German and Swedish datasets from the CONLL-X Shared Task, i.e. the TIGER treebank [21] and Talbanken05 [22]. The TIGER treebank contains 700,000 tokens or 39,200 sentences, and Talbanken05 contains 191,000 tokens or 11,000 sentences. We use the official train-test splits with one important exception: we use the first approx. 5,000 lines in the training data as development data.

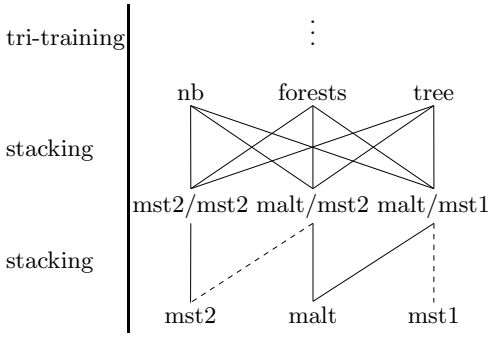
The unlabeled data were the Leipzig Corpora Collection corpora for German and Swedish, except we only used the first 75.000 sentences for the German corpus.

3.2 POS Tags

The unlabeled data were POS tagged using the freely available SVMTool [23] (model 4, left-right-left). The German data set contains 52 POS tags, and the Swedish data set contains 37 POS tags. The accuracy of SVMTool on the two data sets is about 95%.

3.3 Algorithm

Once our data has been prepared, we train the stacked dependency parsers listed in Sect. 2.3 and use them to parse our development data, our test data and our unlabeled data. This gives us three sets of predictions for each of the three data sets. From each triad of predictions we construct two data sets: one for our attachment classifier, say data set A, and one for our dependency labeler, say data set B. Using 5-fold cross validation we train three classifiers on the development (test) data in A and B. The entire architecture can be depicted as follows:



We first stack three dependency parsers as described in Martins et al. [1]. We then stack three classifiers on top of these dependency parsers (and POS tags): a naïve Bayes classifier, random forests, and a decision tree. We select a stopping criterion for the semi-supervised learning of our attachment variable

on development data and unlabeled data in A, and a stopping criterion for our dependency label variable on development data and unlabeled data in B. The stopping criteria are used when we update the classifiers trained on our test data in 5-fold cross validation.

3.4 Baselines

The best of the stacked input parsers is of course our natural baseline. Since we have generalized tri-training, we include the original tri-training algorithm as a semi-supervised baseline. The original tri-training algorithm is run with the same decomposition and the same features as our generalized tri-training algorithm. We use the two learning algorithms of the three originally used in Li and Zhou [10] that we had available, namely naive Bayes and C4.5. Finally, we include S3VMs as a semi-supervised baseline. Since S3VMs produce binary classifiers, and one-vs.-many combination would be very time-consuming, we train a binary classifier that produces a probability that any candidate arc is correct and do greedy head selection. We optimized the feature set and included a total of seven features (head POS, dependent POS, dependent left neighbor POS, distance+direction, predictions of the three classifiers).

4 Results

Our results on the German and Swedish data sets are presented in Figure 2.

We first list the individual parsers (malt and mst2) and the stacked parsers. Since Martins et al. [1] found that for German the level 0 parser mst2 outperformed any configuration of mst2/mst2, there is no figure for mst2/mst2 for German. The best input parser for German is mst2, i.e. the second order MSTParser, whereas the best input parser for Swedish is malt/mst2, i.e. the second order MSTParser stacked on MaltParser with feature configuration D (see Sect. 2.3).

Generalized tri-training leads to highly significant improvements on both data sets ($p < 0.001$). The row "tri-training" lists the results of the initial greedy head selection, whereas tri-training-MST lists the results of reparsing using CLE to produce well-formed dependency trees. Reparsing hurts labeled attachment score (LAS) a bit, but differences are small. Since using the Eisner algorithm for reparsing [24] led to a slightly better result for Swedish than using CLE, we report both results.

We only applied our semi-supervised baselines to unlabeled parsing, but it is quite evident that these learning strategies seem less promising than generalized tri-training. S3VMs seem to average out the input parsers for German, and lead to a relatively small ($< 0.5\%$) improvement for Swedish. The original tri-training algorithm leads to scores well below any of the input parsers for Swedish, but to a considerable improvement for German.

German	LAS	UAS	LA	Δ LAS	p-value
malt	80.08%	82.70%	88.02%		
mst2	84.25%	87.20%	91.38%		
malt/mst2	81.40%	84.18%	89.94%		
malt/mst1	81.35%	84.16%	90.06%		
s3vms	-	84.51%	-		
orig-tri-training(nb)	-	89.06%	-		
orig-tri-training(C4.5)	-	89.22%	-		
tri-training	**85.97%**	**90.24%**	**92.55%**	1.72%	0.0005
tri-training-MST	85.88%	90.13%	92.55%	1.61%	0.0009
tri-training (excl. punct.)	85.84%	90.69%	91.53%		
Martins et al. (2008) (excl. punct.)	*87.44%	-	-		

Swedish	LAS	UAS	LA	Δ LAS	p-value
malt	83.29%	87.52%	87.54%		
mst2	81.95%	87.46%	87.41%		
mst2/mst2	82.32%	87.98%	87.20%		
malt/mst2	83.50%	87.96%	88.24%		
malt/mst1	83.45%	87.87%	88.12%		
s3vms	-	88.45%	-		
org-tri-training(nb)	-	87.22%	-		
org-tri-training(C4.5)	-	87.36%	-		
tri-training	**85.86%**	**91.32%**	**90.12%**	2.36%	0.0001
tri-training-MST	85.71%	91.16%	90.12%	2.21%	0.0001
tri-training-Eisner	85.79%	91.09%	90.12%	2.29%	0.0001
tri-training (excl. punct.)	85.94%	91.95%	89.15%		
Martins et al. (2008) (excl. punct.)	*85.16%	-	-		

Fig. 2. Results on the German and Swedish data sets. Scores are **including punctuation** unless otherwise noted. Δ and p-value is difference with respect to best input parser. *The results in Martins et al. (2008) are incomparable, since they did not take out 5000 sentences for development, and since we did not convert the treebanks into projective trees before training MaltParser.

5 Error Analysis and Discussion

Our error reductions in LAS over the best of our stacked input parsers are 11.16% for German and 12.01% for Swedish; in unlabeled attachment score (UAS), it is 21.42%, resp. 19.55%. The most striking difference between our errors and those committed by our best input parser is their distribution across dependency length, as illustrated in Figure 3. Inference from large amounts of unlabeled data seems to make our parser much better at predicting and labeling long distance dependencies and dependencies of the root node. In general generalized tri-training improves LASs by 1.72% (German) and 2.36% (Swedish), but if we limit attention to long distance dependencies (≥ 7), increases in F-score are even more dramatic, i.e. 5.09% (German) and 8.58% (Swedish).

German		Baseline			System			
length	tokens	rec	prec	F-score	rec	prec	F-score	Δ
root	357	96.92%	96.92%	96.92%	98.04%	97.22%	97.63%	0.71%
1	2295	94.55%	93.49%	94.02%	95.90%	93.42%	94.64%	0.62%
2	964	91.18%	88.52%	89.83%	92.53%	89.83%	91.16%	1.33%
3–6	1233	85.97%	85.83%	85.90%	87.35%	90.05%	88.99%	3.09%
7–	845	83.20%	89.21%	86.10%	88.17%	94.42%	91.19%	**5.09%**
Swedish		Baseline			System			
length	tokens	rec	prec	F-score	rec	prec	F-score	Δ
root	389	91.50%	91.50%	91.50%	95.37%	94.88%	95.12%	3.62%
1	2705	93.83%	94.31%	94.07%	96.23%	94.72%	95.47%	1.40%
2	1133	92.85%	90.69%	91.76%	94.44%	91.45%	92.92%	1.16%
3–6	929	81.81%	83.33%	82.56%	85.79%	88.46%	87.10%	4.54%
7–	500	81.60%	80.95%	81.27%	85.00%	95.29%	89.85%	**8.58%**

Fig. 3. Recall, precision and F-score binned on dependency length. Number of tokens is number of dependencies of length n in the gold standard. Scores are **including punctuation**. Δ is the difference between system and baseline F-scores.

Our errors and those committed by our best input parser seem to be distributed over POS tags in much the same way. Distributions over dependency labels do shed some more light on what kind of long distance dependencies our parsers learn to recover.

Consider the recall, precision and F-score of labeled attachment of the 10 most frequent dependency relations in German, excl. ROOT, in Figure 4 and the same results for the 10 most frequent dependency relations in Swedish, excl. ROOT, in Figure 5.

It is evident that major improvements are primarily due to improvements with coordinating conjunctions, punctuations and complements that can move relatively freely in and across clauses, e.g. subjects, clausal objects and other objects. In Swedish, adverbs and some postnominal modifiers move rather freely, and we see big improvements here as well.

Since we train on less material than Martins et al. [1], taking out 5000 sentences for development, our point of departure is significantly worse than theirs. The best stacking configuration for Swedish, i.e. stacking the second-order MSTParser on MaltParser with features D, has a LAS of 85.16 on the full training section with projectivization and deprojectivization, but only a LAS of 83.93 on our smaller subset (excl. punct.). Consequently, the fact that tri-training leads to results considerably better than those reported in Martins et al. [1] ($\Delta = 0.78$ excl. punct.) says something about the potential of inference from unlabeled data.

Since we greedily select the best head for each word, our output is not guaranteed to be wellformed dependency trees. This is similar to Zeman and Zabokrtsky [25]. The percentage of cyclic structures produced by our ensemble is in both cases below 5%. Surdeanu and Manning [9] observe similar figures in ensemble-based greedy head selection. Reparsing only leads to a small decrease in LAS.

		Baseline			System			
tag	tok. meaning	rec	prec	F-score	rec	prec	F-score	Δ
AG	150 genitive attribute	73.33%	76.39%	74.83%	82.00%	72.35%	76.87%	2.04%
CD	129 coord. conjunction	73.64%	71.97%	72.80%	75.19%	74.62%	74.90%	2.10%
CJ	172 conjunct	67.44%	67.05%	67.24%	69.19%	67.61%	68.39%	1.15%
MNR	153 postnominal mod.	58.17%	54.27%	56.15%	58.82%	54.55%	56.60%	0.45%
MO	772 modifier	72.28%	74.40%	73.32%	76.55%	76.16%	76.35%	3.03%
NK	1721 noun kernel mod.	96.22%	95.61%	95.91%	95.64%	96.31%	95.97%	0.06%
OA	206 accusative object	74.27%	72.17%	73.20%	73.30%	72.25%	72.77%	-0.43%
OC	220 clausal object	90.45%	85.04%	87.66%	92.27%	89.82%	91.03%	**3.37%**
PUNC	808 punctuation	84.28%	84.28%	84.28%	86.76%	86.65%	86.70%	2.42%
SB	425 subject	86.35%	82.84%	84.56%	87.53%	88.15%	87.84%	3.28%

Fig. 4. Recall, precision and F-score of labeled attachment score of 10 most frequent dependency relations in German

		Baseline			System			
tag	tok. meaning	rec	prec	F-score	rec	prec	F-score	Δ
++	184 coord. conjunction	91.30%	89.84%	90.56%	95.65%	94.62%	95.13%	4.57%
AA	266 other adverbial	62.41%	64.59%	63.48%	67.67%	67.92%	67.79%	4.31%
AT	234 nom. pre-modifier	96.15%	96.15%	96.15%	96.58%	95.76%	96.17%	0.02%
CC	220 conjuncts	79.09%	79.82%	79.45%	82.27%	83.80%	83.03%	3.58%
DT	556 determiner	94.96%	90.26%	92.55%	95.32%	92.66%	93.97%	1.42%
ET	342 other nom. postmod.	72.22%	73.73%	72.97%	82.16%	79.15%	80.63%	7.66%
IP	312 period	91.35%	91.35%	91.35%	100.00%	100.00%	100.00%	**8.65%**
OO	284 other object	85.92%	78.96%	82.29%	89.08%	84.05%	86.49%	4.20%
PA	677 prep. compl	95.57%	94.04%	94.80%	96.01%	95.73%	95.87%	1.07%
SS	508 other subject	90.55%	89.67%	90.11%	93.70%	91.54%	92.61%	2.50%

Fig. 5. Recall, precision and F-score of labeled attachment score of 10 most frequent dependency relations in Swedish

Generalized tri-training leads to much better results than the other two semi-supervised learning algorithms. The original tri-training algorithm gives a big improvement for German, but for Swedish it makes things much worse. S3VMs do not lead to improvements, but seem to average out the ensemble used in the stacking.

6 Conclusion

This paper showed how the stacked dependency parsers introduced in Martins et al. [1] can be improved by inference from unlabeled data. Briefly put, we stacked three diverse classifiers on triads of stacked dependency parsers and let them label unlabeled data for each other in a co-training-like architecture. Our error

reductions in LAS over the best of our stacked input parsers were 12.24% for German and 12.01% for Swedish; in UAS, it was 21.42%, resp. 19.55%. Error analysis shows that improvements are primarily due to better recovery of long distance dependencies. Generalized tri-training improves LASs by 1.72% (German) and 2.36% (Swedish), but if we limit attention to long distance dependencies (≥ 7), increases in F-score are even more dramatic, i.e. 5.09% (German) and 8.58% (Swedish).

References

1. Martins, A., Das, D., Smith, N., Xing, E.: Stacking dependency parsers. In: EMNLP, Honolulu, Hawaii (2008)
2. Rimell, L., Clark, S., Steedman, M.: Unbounded dependency recovery for parser evaluation. In: EMNLP, Singapore (2009)
3. Abney, S.: Semi-supervised learning for computational linguistics. Chapman and Hall, Boca Raton (2008)
4. Wolpert, D.: Stacked generalization. Neural Networks 5, 241–259 (1992)
5. Sagae, K., Lavie, A.: Parser combination by reparsing. In: HLT-NAACL, New York City, NY (2006)
6. Hall, J.: colleagues: Single malt or blended? In: CONLL, Prague, Czech Republic (2007)
7. Nivre, J., McDonald, R.: Integrating graph-based and transition-based dependency parsers. In: ACL-HLT, Columbus, Ohio (2008)
8. Fishel, M., Nivre, J.: Voting and stacking in data-driven dependency parsing. In: NODALIDA, Odense, Denmark (2009)
9. Surdeanu, M., Manning, C.: Ensemble models for dependency parsing: cheap and good? In: NAACL, Los Angeles, CA (2010)
10. Li, M., Zhou, Z.H.: Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering 17(11), 1529–1541 (2005)
11. Koo, T., Carreras, X., Collins, M.: Simple semi-supervised dependency parsing. In: ACL, Columbus, Ohio (2008)
12. Wang, Q., Lin, D., Schuurmans, D.: Semi-supervised convex training for dependency parsing. In: ACL, Columbus, Ohio (2008)
13. Suzuki, J., Isozaki, H., Carreras, X., Collins, M.: Semi-supervised convex training for dependency parsing. In: EMNLP, Singapore (2009)
14. Sagae, K., Tsujii, J.: Dependency parsing and domain adaptation with lr models and parser ensembles. In: EMNLP-CONLL, Prague, Czech Republic (2007)
15. Chen, W., Zhang, Y., Isahara, H.: Chinese chunking with tri-training learning. In: Computer processing of oriental languages, pp. 466–473. Springer, Berlin (2006)
16. Nguyen, T., Nguyen, L., Shimazu, A.: Using semi-supervised learning for question classification. Journal of Natural Language Processing 15, 3–21 (2008)
17. Sindhwani, V., Keerthi, S.: Large scale semi-supervised linear SVMs. In: ACM SIGIR, Seattle, WA (2006)
18. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: HLT-EMNLP 2005, Vancouver, British Columbia (2005)
19. Nivre, J.: Colleagues: MaltParser. Natural Language Engineering 13(2), 95–135 (2007)

20. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
21. Brants, S., Hansen, S., Lezius, W., Smith, G.: The TIGER treebank. In: TLT, Sozopol, Bulgaria (2002)
22. Nilsson, J., Hall, J., Nivre, J.: MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity. In: NODALIDA, Joensuu, Finland (2005)
23. Gimenez, J., Marquez, L.: SVMTool: a general POS tagger generator based on support vector machines. In: LREC, Lisbon, Portugal (2004)
24. Eisner, J.: Three new probabilistic models for dependency parsing. In: COLING, Copenhagen, Denmark (1996)
25. Zeman, D., Žabokrtský, Z.: Improving parsing accuracy by combining diverse dependency parsers. In: IWPT, Vancouver, Canada (2005)

Shooting at Flies in the Dark: Rule-Based Lexical Selection for a Minority Language Pair

Linda Wiechetek[1], Francis M. Tyers[2], and Thomas Omma[3]

[1] Giellatekno
Romssa Universitehta, Norway
linda.wiechetek@uit.no
[2] Dept. Lleng. i Sist. Inform., Universitat d'Alacant, Spain
ftyers@dlsi.ua.es
[3] Divvun, Sámi Parliament, Norway
thomas.omma@uit.no

Abstract. This paper presents a set of rules which form the prototype lexical selection component of a rule-based machine translation system between two closely-related minority languages, North Sámi and Lule Sámi. While the languages have comprehensive monolingual computational linguistic resources, they lack bilingual resources. One-to-one relations in the lexicon dominate, but there are also more complex relations that require lexical selection using both lexical and syntactico-semantic context. An evaluation was performed over a set of 11 word pairs, which shows that constructing lexical selection rules and doing research on a North Sámi–Lule Sámi contrastive lexicon is an interrelated process. Other lesser-resourced language pairs will benefit from the use of lexical selection rules as the relevance of lexical selection increases with the divergence of the languages.

Keywords: Sámi languages, rule-based lexical selection, MT.

1 Introduction

North Sámi and Lule Sámi belong to Sámi group of languages which is a subfamily of the Finno-Ugric language family. They are spoken in the north of the Nordic countries. North Sámi has between 15,000 and 25,000 speakers and Lule Sámi has around 2,000 speakers.

The languages are neighbours and are mutually intelligible, although often the majority language is used as a *lingua franca*. Despite their mutual intelligibility, orthographic differences impede reading comprehension between the two languages.

A large amount of word roots and morphosyntactic categories (cases, inflectional and derivational patterns etc.) are shared between the languages. However, despite the relatedness, there are a number of challenges in machine translation for the two languages.

H. Loftsson, E. Rögnvaldsson, S. Helgadóttir (Eds.): IceTAL 2010, LNAI 6233, pp. 418–429, 2010.

Both languages have phonemic orthographies, with differences resulting from differing conventions in standardisation. These differences can largely be handled by rules, thus a bilingual dictionary between the two was created from scratch by simply converting the orthography.

This process provides an adequate lexicon, but when inspecting the resulting translations with a native Lule Sámi speaker, the situation was found to not be as simple as originally assumed.

On the syntactic level, there are obvious differences, such as an asymmetry in the case system (North Sámi locative being expressed by Lule Sámi elative and inessive) and differences in word order, Lule Sámi tends towards an OV word order, where North Sámi tends towards VO.

Other differences are less obvious, while North Sámi can express the semantic notion of path with both a *-ráigge* 'along' compound construction[1] (1-a) and a genitive case adverbial (1-b), Lule Sámi (1-c) lacks the simple genitive construction. Therefore, the translation of(1-a) is .

(1) a. Soai boahtiba geainnoráigge. (North Sámi)
They-DU come this way-along
'They come along the way.'

b. Soai boahtiba dán geainnu. (North Sámi)
They-DU come this way-GEN
'They come along this way.'

c. Såj boahteba gæjnnorájge. (Lule Sámi)
They-DU come this way-along
'They come along this way.'

The languages also diverge on the lexical level, and although the automatically constructed bilingual lexicon often provides adequate translations, in many cases word use is actually quite different. In some cases historical word roots are different, in other cases, words in one of the languages have acquired a new sense which does not exist in the other language or appear in specific syntactic or semantic construction which is resolved differently in the other language.

The need for lexical selection[2] came up when seemingly straightforward translations were not accepted by Lule Sámi native speakers. The errors could not be fixed by correcting the bilingual dictionary because the translation error lies not in any erroneous entries, but in the one-dimensionality of the entries. Less related languages will profit from lexical selection to an even larger extent as natuarlly semantic concepts will diverge more the further languages and the speech communities differ from each other.

[1] The latter part of the lexicalised construction is originally a genitive too.

[2] Lexical selection is defined by [1] as the "principled selection of a) lexical items and b) the syntactic structure for input constituents, based on lexical semantic, pragmatic and discourse clues available in the input."

2 Objectives

The objectives behind the development of a machine translation (MT) system between the two languages are largely guided by the sociolinguistic situation.

Following [2], applications of machine translation can be divided in two main groups with different requirements: *assimilation*, that is, to enable a user to understand what the text is about; and *dissemination*, that is, to help in the task of translating a text to be published.

Assimilation may be possible even when the text is far from being grammatically correct; however, for dissemination, the effort needed to correct (*post-edit*) the text must be lower than the effort needed to translate it from scratch.

A majority to minority language system will mainly be used for dissemination purposes, where post-editing the output should be faster than translating from scratch and intelligibility is less important.

In a minority to majority language system on the other hand, intelligibility is the main goal as MT is mainly used for assimilation, for instance, to answer vital questions such as "what are they writing about me in the minority language newspaper?".

The system described in this paper falls outside the usual majority–minority continuum, as both languages can be considered minority languages (one of which again is a minority language in the Sámi context), and the system has a dual focus.

On one hand, it should be able to produce Lule Sámi texts appropriate for post-edition from North Sámi texts, for example to translate educational materials.

On the other hand, it should also be useful for assimilation, to give Lule Sámi speakers the opportunity to follow news in North Sámi (for example from the daily published newspaper *Ávvir*[3]).

3 Technical Background

This section gives a brief overview of the two main technologies used in the construction of the prototype system,[4] Apertium,[5] a rule-based machine translation platform, and Constraint Grammar, [3] a rule-based framework for the disambiguation and annotation of text.

3.1 Apertium

The Apertium platform was originally aimed at the Romance languages of the Iberian peninsula, but has also been adapted for other language pairs, such

[3] http://avvir.no

[4] The prototype system may be tested online at:
http://victorio.uit.no/cgi-bin/francis/index.php

[5] http://www.apertium.org

as Welsh [4] and Basque [5]. The whole platform, both programs and data, is available from the project website under the GPL licence.[6]

The engine largely follows a shallow-transfer approach to machine translation [6]. Finite-state transducers [7] are used for lexical processing, first-order hidden Markov models (HMM) and optional Constraint Grammar are used for part-of-speech tagging, and finally multi-stage finite-state based chunking is used for structural transfer.

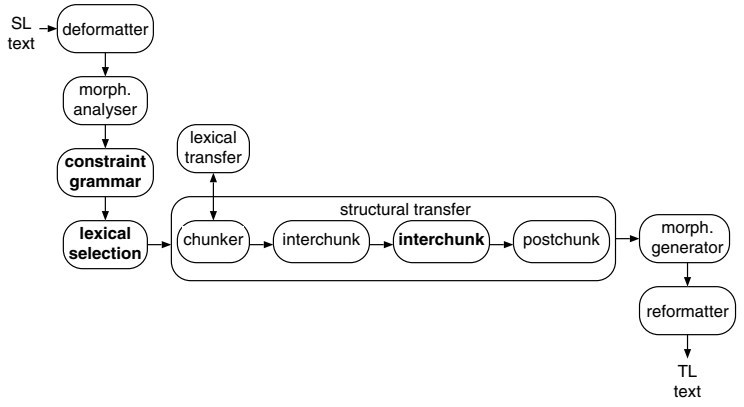


Fig. 1. Modular architecture of the Apertium MT platform. Bold indicates adjustments made for the North Sámi to Lule Sámi pair. The HMM-based part-of-speech tagger has been replaced with a constraint grammar which also provides syntactic labelling, a lexical selection module, also based on a constraint grammar has been inserted, along with an extra *interchunk* (structural transfer) module to deal with longer distance reordering. For example reordering co-ordinated noun-phrase objects, in the first stage of transfer, the co-ordinator and the noun phrases are chunked, then in the first interchunk stage, the two noun phrases are chunked together with the co-ordinator, and finally in the second interchunk stage, they are moved as one unit.

As this paper focuses on the lexical selection aspect, a more detailed description of the pipeline (figure 1) will not be made.

3.2 Constraint Grammar

The formalism used for both disambiguation and annotation is Constraint Grammar, which is a linguistically-based approach used for the bottom-up analysis of running text. The Sámi Constraint Grammar performs both morphological and syntactic disambiguation, and annotates syntactic and dependency labels. It uses the VISL-CG3[7] implementation that has been further developed to include support for annotating dependency relations.

[6] http://www.fsf.org/licensing/licenses/gpl.html
[7] http://visl.sdu.dk/constraint_grammar.html

The lexical selection module is also implemented in Constraint Grammar, and annotates words which are ambiguous in translation in a disambiguated source language sentence with references to their translation in the target language.

This method is inspired by other MT systems including rule-based lexical selection, such as the *Dan2Eng* system [8] which successfully uses 17,000 hand-written lexical transfer rules.

4 Lexical Selection

4.1 Potential Candidates

The bilingual North Sámi–Lule Sámi lexicon contains many one-to-one relations such as *eadni* → *ieddne* 'mother', and many nouns seem to have fairly straightforward translations.

Upon closer inspection one-to-many, many-to-many, and many-to-one relations are found in the bilingual dictionary. However, many-to-one translations will not be dealt with in this paper as the translation currently is limited to the direction North Sámi→Lule Sámi. Some lexical entries are polysemous in both directions. The North Sámi verb *ráhkadit* 'make' translates into *dahkat* 'make, do' and *stiellit* 'prepare'. Lule Sámi *dahkat* on the other hand translates both into North Sámi *ráhkadit* and *dahkat* 'do'. The current set of one-to-many wordpairs includes verbs, nouns, adjectives and adverbs. Any word with two or more (unrelated) meanings (homonymy or polysemy) is a candidate for lexical selection.

In both languages there is substantial homonymy and polysemy between inflected forms of words, but little between lemmata, although it does exist, for example, *luohkká* 'hill' or 'class, grade' and *giella* 'language, snare, lasso-ring'. In some cases, traditional words which have acquired a modern meaning, e.g. North Sámi *cuozza* 'skin/membrane, transparency', originally only 'membrane'. This polysemy is partly preserved in Lule Sámi (as in *giella*). In other cases, Lule Sámi uses different words for the different senses of the North Sámi noun (as in *luohkká*).

Some words have acquired a metaphorical sense *jámas* 'dead' as in *jámas dolkan* 'dead sick of' – possibly under the influence of the Scandinavian languages, i.e. *dødslei* 'dead-sick.of' in Norwegian.

In other cases, words that can be used in a wider, narrower or otherwise different domain need to have lexical selection rules written, for example the adjective *boaris* 'old' which can only be translated as *boares* for inanimate objects, and otherwise translates as *vuoras*.

4.2 Rules

Within the Constraint Grammar, semantic information is encoded in semantic sets within the lexical selection module. The bilingual lexicon specifies one or more alternative translations, the default labelled with `s0`, and the alternatives

labelled with consecutive numbers from one. The rules make use of morphological, syntactic and semantic information. Rules were inspired by comments by a native speaker of Lule Sámi about incorrect lexical choice in the translations made by the MT system. A number of word-pairs with context-dependent translations were identified. A native speaker of North Sámi with passive knowledge of Lule Sámi was asked to translate a number of Lule Sámi sentences including the different variants so that the contexts for each translation variant could be refined.

The example in (2) shows how PoS/morphological information can be used to create a rule to distinguish the translations of *luohkká* 'hill, class or grade', where the sense *klássa* 'class or grade' is used with a preceeding ordinal. The other translation *luohkka* 'hill' on the other hand is less likely to be the correct one if the word is preceeded by an ordinal.

(2) Sii leat vuosttaš luohkás. (North Sámi)
They are first grade-LOC
'They are in first grade.'

The rule selecting the translation *vuoras* for *boaris* 'old' makes use of the fact that personal pronouns in first and second person usually denote a human. Syntactic information is specifically used with the polysemous verb *orrut* 'stay, seem', which translates into *vuojnnet* before a noun/adjective in essive case[8] or a predicative, as in example (3).

(3) Orru leamen buorre. (North Sámi)
Seem be-ACTIO.ESS good-PRED
'It seems to be good'

The last type of constraints are narrower lexical or even idiosyncratic constructions. The adjective *buorre* 'good' is translated into *jasskat* before particular nouns, such as *iešdovdu* 'self-confidence'. Example (4) shows an example of this kind of constraint, in the example North Sámi is given on the first line and Lule Sámi on the second line.

(4) ... addin dihte *buori* ieš̆dovddu. (North Sámi)
... vattátjit *jasska* iesjdåbdov. (Lule Sámi)
'... in giving good self-esteem'

Semantic information is used in a number of rules. The noun *luohkká* 'hill, class' is translated into *klássa* 'class' in a sentence containing members of a set of words related to education. The verb *ráhkadit* 'make, prepare' is translated into *stiellit* in sentences that have a grammatical object from a set of words related to food. The rule translating *boaris* 'old' into *vuoras* makes use of a set generalising over nouns denoting humans, see figure 2. The adverb *jámas* 'dead' translates into *sælldát* in connection with *psych-verbs*,[9] for example *ballat* 'fear' *dolkat* 'be sick

[8] The essive case expresses a temporary state or quality.
[9] Psych-verbs are those verbs which designate a psychological state or process.

of' *suhttat* 'get angry at', where it gets a metaphorical meaning which is not conveyed by the word *jámas* in Lule Sámi.

Lexical selection can be handled by rules that pick a certain sense of a word. This sense is chosen in a certain syntactic or semantic context and then receives a particular translation in the target language.

North Sámi *boaris* 'old' can be translated with both *vuoras* and *boares*. *vuoras* is used only for humans and animals. The set ANIMAL includes nouns that denote animals such as *ealga* 'elk' and *rievssat* 'ptarmigan'. The set HUMAN includes male, female proper nouns, surnames and other nouns denoting humans such as *áddjá* 'grandfather' and *oahpaheaddji* 'teacher'. The rule in 2 selects sense 1 `(S1)` *boares* when the adjective `(A)` has an attributive form `(A Attr)`, and the noun `(N)` it modifies is not in the set of nouns that denote humans or animals `(LINK NOT 0 HUMAN OR ANIMAL)`.

```
SET ANIMAL = "ealga" "rievssat" ...;
SET HUMAN  = (Prop Mal) (Prop Fem) (Prop Sur)
             "áddjá" "oahpaheaddji" ... ;

SUBSTITUTE (A S0) (A S1) ("boaris"ri A Attr)
           (*1 N BARRIER NOT-Attr LINK NOT 0 HUMAN OR ANIMAL);

SUBSTITUTE (IV) (IV S1) ("orrut"ri V) (1 (@←SPRED));
```

Fig. 2. Two constraint grammar lexical selection rules to select between two translations of *boaris* 'old' and *orrut* 'stay, seem'

The second rule selects sense 1 `(S1)` of the intransitive verb `(IV)` *orrut* 'stay, seem' if there is a subject predicative `(@←SPRED)` one position to the right as in example .

(5) a. Orru buorre. (North Sámi)
Seems good.
'It seems good.'

b. Årru buorre. (Lule Sámi)
Seems good.
'It seems good.'

5 Evaluation

For the evaluation, the North Sámi side of the New Testament was tagged and sentences with the target words were extracted.[10] Both equivalent and

[10] The corpus of test sentences may be downloaded from:
http://www.dlsi.ua.es/~ftyers/sme-smj.testsentences.tar.gz

non-equivalent translations were considered, but only equivalent translations were included when calculating the percentage of correct translations.

Equivalent constructions are those where the lexical item is translated by a possible equivalent of the same part-of-speech. Derivations which do not change the lexical category of the word (e.g. Noun → Noun) and compounds are permitted. In some cases it was difficult to decide whether the translation is a possible equivalent or a different word. As is the case with the North Sámi verb *muitalit* 'tell'. Non-equivalent constructions are those where the lemmata in question are not possible lexical equivalents, the syntactic construction differs completely or the lexical equivalent is simply left out, compare the aligned translations in (6) and (7).

(6) Muhto ii son eallán suinna ovttas (North Sámi)
But not he/she live+PP him+COM together
'But she did not live together with him'

(7) Valla ittjij suv duohtada (Lule Sámi)
But not+PRT he/she+ACC touch+CONNEG
'But she did not touch him'

Rather than aiming at a system that translates (6) into (7), the aligned sentence should be discarded in favour of a more literal translation.

Some potential Lule Sámi equivalents have better North Sámi equivalents than *muitalit* 'tell' and it is questionable in how far these can be considered equivalent constructions: Lule Sámi *sárnnot* is usually translated with *dadjat* 'say, tell' in North Sámi and *hållat* with *hupmat* 'talk, speak' or *hállat* 'talk, speak'.

These sentences were run through the rules and the chosen translation was checked against the translation in the Lule Sámi New Testament. The difficulty lies in finding appropriate text for the evaluation. The number of parallel texts for North Sámi and Lule Sámi is very limited. For evaluation the New Testament is used, which exists as parallel text for North and Lule Sámi, but is based on different originals, in different languages. That means that the closest aligned sentences do not neccessarily contain lexical equivalents, possibly not even corresponding syntactic constructions.

A number of other problems can also be predicted: the Biblical language might not catch the newer (metaphorical) senses of a word and in general not reflect the language to which the MT system is targetted and the New Testament might miss out contexts in which word senses could be used for different reasons.

What is more, Lule Sámi does not have a strict norm (as opposed to North Sámi). Lexical dialect borders are fuzzy, which makes it even harder to decide if different word choice is simply due to dialect variation with a use restricted to a dialect context or words that can be distinguished in their use by means of linguistic constraints.

Table 1. Evaluation of the lexical selection rules over the New Testament. The first column gives the word in North Sámi, and the fourth column the possible translations into Lule Sámi from the bilingual lexicon with the default underlined. The *Type* column shows the type of rule (lexical, word category, syntactic, semantic). The *Equiv.* column gives the number of equivalent sentences which were found for the word in both the North and Lule Sámi New Testaments, while the *Correct* column gives the number of correct translations produced by the rules.

Word	Gloss	Type	Translations	Type	Equiv.	Correct	(%)
jámas	'dead'	Adv.	jámas, sælldát	Sem	2	2	100%
láhkái	'kind, type'	Adv.	láhkáj, muoduk	Synt	2	2	100%
čeahppi	'smart'	Adjec.	smidá, tjiehppe	Lex	2	2	100%
buorre	'good'	Adjec.	buorre, jasskat	Lex	218	192	88%
eallit	'to live'	Verb	viessot, iellet	Lex	147	115	78%
orrut	'to stay, to seem'	Verb	årrot, vuojnnet	Synt	87	50	57%
ráhkadit	'to make'	Verb	dahkat, stiellit	Sem/Synt	33	16	48%
muitalit	'to tell'	Verb	subtsastit, mujttalit	Lex	102	28	27%
boaris	'old'	Adjec.	vuoras, boares	Sem/Synt	31	8	25%
hui	'very'	Adv.	huj, sieldes	Synt	8	0	0%
jaska	'quiet'	Adv.	sjávot, jasska	Lex	0	0	-
luohkká	'class, hill'	Noun	luohkka, klássa	Sem/Cat	0	0	-

6 Discussion

As can be seen in table 1, the rules perform best on words where the non-default scope is quite narrow, such as the rule for *buorre* 'good', where the non-default is only picked in some lexical contexts. Bad performance of some of the other rules is due to the selection of the wrong default as e.g. in the case of *boaris* 'old', the existence of several variants that have not been considered (and might be even restricted to a Biblical context), as in *muitalit* 'tell', where *giehttot* and not *subtsastit* or *mujttalit* get most hits, and difficulties in excluding synonymy.

It is also due to the inclusion of various potentially deviating contexts in the total number of equivalent sentences as in the case of *muitalit* 'tell'. Categorising the rules with regard to their linguistic level and complexity, the simple lexical rules (referring to nearly idiosyncratic contexts) are written very quickly and make up the ones performing best (with the exception of the rule for *láhkái* 'kind, type' and *jámas* 'dead', which are hard to evaluate since each of them only occurs twice.). In table 1, most of the syntactico-semantic rules perform worst. The higher the level of abstraction, either in terms of semantic sets or syntactic contexts, the more carefully the rules need to be made to achieve good performance. The higher the level of abstraction is, the better one needs to know positive and negative contexts of the word on the one hand and all the possible equivalents of a word on the other hand.

The evaluation has helped to specify the contexts of lexical selection in some cases. The rule selecting *iellet* as a translation for *eallit* 'live, be alive' originally had a very narrow context, which now can be extended to other contexts than

agálaččat 'forever'. The verb *iellet* is translated by [9] as *leva (vanligen i andlig betydelse)* 'live/be alive (usually in a spiritual sense)'. Typical and recurring constructions for the religious sense of *eallit* are *ealli čáhci* 'living water', *ealli Ipmil* 'living God', and *ealli sátni* 'living word'. The adjective *vuoras* is typically selected as a translation for *boaris* 'old' after the word *jahki* 'year'.

Secondly, we found additional equivalents for North Sámi words. The verb *muitalit* 'tell' has been translated as *subtsastit*, *mujttalit*, *giehttot*, *sárnnot*, *ságastit*, *javllat*, *hållat*, *diededit*. The verb *ráhkadit* has been translated as *dahkat*, *stiellit*, *tsieggit*, *gárvedit*. The adjective *boaris* 'old' does not only have the translations *boares* and *vuoras*, but also *oames* in contrastive phrases about *ådå* 'new', *varás* 'fresh' and *oames* 'old', in food contexts, and about clothes 'worn out'. Other than in SMT, in RBMT, one (most) suitable translation can be picked generalizing over the variation that can be found in parallel texts.

The improved rule set picks out *oames* if the sentence contains either *ådå* 'new' or *varás* 'fresh', and *boaris* 'old' stands in attributive position to a noun from the semantic set of clothes or food.

The rule for the verb *orrut* 'stay, seem' performs well in selecting the non-default *vuojnnet* 'seem', but often gets translated not only with *årrot* 'stay' or *vuojnnet* 'seem', but simply with *liehket* 'be' as in (8-b).[11] The verb *liehket* has a closer equivalent in North Sámi, *leat* 'be'. The goal of the machine translation system is not to get caught up in possible correct variants. Linguistically on the other hand this finding is interesting, as it could hint at a more frequent use of *liehket* in Lule Sámi than in North Sámi.

(8) a. Ehpet go dieđe [...] ahte Ipmila Vuoigŋa orru din siste?
Not ø-qst know [...] that God-gen Spirit stays you inside?
(North Sámi)

'Don't you know that God's Spirit lives in you?'

b. Ehpit gus dádjada [...] Jubmela Vuojŋŋanis dijájn le?
Not ø-qst know [...] God-gen Spirit you-LOC PL stays?
(Lule Sámi)

'Don't you know that God's Spirit is in you?'

In some cases the default needs to be reconsidered. The adjective *boaris* 'old' is translated more frequently as *boares* than as *vuoras* (as originally predicted). A bigger corpus might be needed to test and compare the frequency of non-human vs. human use of the word.

Some of the word pairs seem to be very difficult to distinguish and have rather a synonymy status as even seemingly clear contexts included both variants (*eallit agálaččat* both *iellet* and (a few) *viessot*). In other cases it is more important to distinguish (*orrut*, *boaris*).

[11] 1 Corinthians 3:16

The New Testament examples showed that even in a seemingly straightforward word pair, the realisation in text can diverge in both directions. This may result in several alternative translations, partly synonymous.

Writing lexical selection rules does not only help to pick the correct equivalent, but also to acquire knowlege about the correct equivalent.

Even though North Sámi and Lule Sámi are closely related languages and a large amount of lexical transfer is fairly straightforward, a number of cases require lexical selection. And the variation found in the New Testament shows that preferences with regard to lexical and syntactic constructions can vary substantially. The reasons for that can lie in individual preferences of the translator, but can also be a general linguistic tendency of the language. Two independently translated texts with different source languages do not provide this information. A lot of native speaker competence and a more bilingual corpus material is needed in further work.

In a non-standardised language like Lule Sámi there is much lexical variation. This makes it difficult to match parallel texts, as texts show no consensus as to which term is the appropriate translation. For each translation one equivalent must be chosen, and here, RBMT may do just that.

Simple and accurate rules, if possible with a high level of abstraction, can improve the output of MT on a lexical level considerably, even between closely-related languages. A rule-based approach to lexical selection also has further benefits where the languages in question are under-studied as it provides an opportunity to do research into lexicography and semantics in both languages.

7 Conclusion

The paper has explored the use of lexical selection in machine translation to improve lexical choice in translation. In the case of such little researched language pairs as North Sámi–Lule Sámi, this does not only include technical work, but also linguistic pioneer work as one cannot base decisions on an existing bilingual dictionary, but rather write parts of the bilingual dictionary oneself. Both linguistic research and technical solutions for lexical selection in machine translation between related under-resourced languages are needed. But of course, lexical selection is necessary to a much larger extent for less related languages. The current approach shows that despite (linguistic) difficulties, the inclusion of a lexical selection module based on Constraint Grammar rules can be included in a fairly simple manner into a rule-based machine translation system such as Apertium.

Acknowledgements. Many thanks to Trond Trosterud, Lene Antonsen and the anonymous reviewers for their helpful comments in improving this paper. This work has also received the support of the Spanish Ministry of Science and Innovation through project TIN2009-14009-C02-01.

References

1. Pustejovsky, J., Nirenburg, S.: Lexical selection in the process of language generation. In: Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics, Morristown, NJ, USA, pp. 201–206. Association for Computational Linguistics (1987)
2. Gachot, D.: Assimilation or dissemination? that is the question. In: Proceedings of the Second Conference of the Association for Machine Translation in the Americas, Montreal, Quebec, Canada (1996)
3. Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A.: Constraint Grammar: A language independent system for parsing unrestricted text. Mouton de Gruyter, Berlin (1994)
4. Tyers, F.M., Donnelly, K.: apertium-cy: A collaboratively-developed free RBMT system for Welsh to English. Prague Bulletin of Mathematical Linguistics (91), 57–66 (2009)
5. Ginestí-Rosell, M., Ramírez-Sánchez, G., Ortiz-Rojas, S., Tyers, F.M., Forcada, M.L.: Development of a free Basque to Spanish machine translation system. Procesamiento del Lenguaje Natural 43, 187–195 (2009)
6. Forcada, M.L., Tyers, F.M., Ramírez-Sánchez, G.: The free/open-source machine translation platform Apertium: Five years on. In: Tyers, F.M., Pérez-Ortiz, J.A., Sánchez-Martínez, F. (eds.) Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation Free RBMT 2009, November 2009, pp. 3–10 (2009)
7. Roche, E., Schabes, Y.: Introduction. In: Roche, E., Schabes, Y. (eds.) Finite-State Language Processing, pp. 1–65. MIT Press, Cambridge (1997)
8. Bick, E.: Dan2eng: Wide-Coverage Danish-English Machine Translation. In: Proceedings of Machine Translation Summit XI, Copenhagen, September 10-14, pp. 37–43 (2007)
9. Korhonen, O.: Báhkogirjje: julevusámes dárrui dáros julevusábmái. Jokkmokk, Sámij åhdadusguovdásj. Dictionary: Lulesámi to Norwegian, Norwegian to Lulesámi (2007)

Author Index