Computational Linguistics in the Netherlands
12 January 2007, Leuven

LOT
Janskerkhof 13
3512 BL Utrecht
The Netherlands

phone: +31 30 253 6006
fax: +31 30 253 6406
e-mail: lot@let.uu.nl
http://www.lotschool.nl

ISBN 978-90-78328-41-4
NUR 616

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands

(CLIN17)

Edited by:

Peter Dirix
Ineke Schuurman
Vincent Vandeghinste
Frank Van Eynde

Centre for Computational Linguistics
Katholieke Universiteit Leuven

LOT
Utrecht
2007

Contents

Preface

This volume is a collection of selected papers from the 17th Meeting of Computational Linguistics in the Netherlands (CLIN-17). It was held at the University of Leuven on January 12, 2007 and organized by the Centre of Computational Linguistics. There were 110 registered participants, mostly from the Netherlands and the Dutch speaking part of Belgium, but the international appeal of CLIN was confirmed once again by the presence of 30 participants from other countries, many of whom also gave a presentation. Another trend is the growing number of participants from companies and institutes of higher education, especially translator schools.

The number of presentations was higher than ever: there were 52 oral presentations, organized in four parallel sessions, and 12 poster presentations. Icing on the cake was the invited lecture on 'Efficient HPSG realization for precision machine translation' by professor Stephan Oepen (University of Oslo, NTNU Trondheim and CSLI Stanford).

In response to the call for papers that was issued after the meeting we received 26 submissions out of which 15 were selected for publication. Since every submission was reviewed by at least two referees we had to mobilize a large group of experts. This went surprisingly smoothly, thanks to the willingness and cooperation of: Gosse Bouma, Michael Carl, Walter Daelemans, Franciska de Jong, Guy De Pauw, Peter Dirix, Jacques Duchateau, Markus Egg, Dirk Heylen, Kris Heylen, Véronique Hoste, Sebastian Kürschner, Sien Moens, Paola Monachesi, John Nerbonne, Pierre Nugues, Nelleke Oostdijk, Hans Paulussen, Martin Reynaert, Ineke Schuurman, Louis ten Bosch, Joerg Tiedemann, Erik Tjong Kim Sang, Vincent Vandeghinste, Antal van den Bosch, Ton van der Wouden, Jan van Eijck, Frank Van Eynde, Hans van Halteren, Gertjan van Noord and Thomas Zastrow.

We also thank the authors of the selected papers for their timely delivery of the final version and for their efforts to comply with our formats and size restrictions. The resulting collection shows a large diversity of topics and approaches, so much so that we have refrained from grouping them into sections and decided to order them alphabetically. Still, if there is one trend, it is the growing number of

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

contributions on matters of semantics and discourse.

For the publication we have followed the example of our Leiden colleagues in 2005 and contacted the Landelijke Onderzoeksschool Taalwetenschap (LOT), proposing to include it in their Occasional Series. Their willingness and their help are hereby gratefully acknowledged.

Gratefully acknowledged are also the contributions of our sponsors: the FWO research network 'Computational Linguistics in Flanders' (CLIF), the research program 'Spraak- en Taaltechnologische Voorzieningen voor het Nederlands' (STEVIN), the Faculty of Arts of the Leuven University, and the research unit of Dutch, German and Computational Linguistics also at the Leuven University. For their help in acquiring these funds we thank Willy Clarysse, Alice Dijkstra, Hans Kruithof, Jan Odijk and Patrick Wambacq.

Finally we wish to thank our predecessors, the organizers of the 16th CLIN meeting in Amsterdam for sharing some of their experiences and for the timely delivery of last year's proceedings, as well as everyone who helped with the local organization.

Leuven, June 2007

Peter Dirix
Ineke Schuurman
Vincent Vandeghinste
Frank Van Eynde

Invited talk: Efficient HPSG Realization for Precision Machine Translation

Stephan Oepen
University of Oslo, NTNU Trondheim, and CSLI Stanford

Abstract

I will review recent advances in grammar-based sentence realization from logical-form meaning representations. The LOGON MT prototype aims at the fully-automated, high-quality translation of Norwegian instructional texts (on backcountry activities) into English. The LOGON generator operates off underspecified meaning representations derived from 'deep' grammatical analysis (in the LFG framework) and subsequent semantic transfer. The generator builds on the LinGO English Resource Grammar (in the HPSG framework) and combines a highly optimized chart-based algorithm with a rich, probabilistic model to rank alternate realizations. Integration of the stochastic model into the enumeration of outputs from the packed chart allows the generator to selectively unpack n-best lists of realizations with minimal search. Besides empirical results for the realization task when evaluated in isolation, I will present a summary of quantitative measures on the current development status (and promise) of the LOGON MT pipeline as a whole.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

1

Identification and handling of dialectal variation with a single grammar

António Branco and Francisco Costa
Universidade de Lisboa

Abstract

We present a study on approaches to handle variation in a deep natural language processing formalism. It allows a grammar to be parameterized as to what language variants it accepts, but also to detect such variants. In this respect, we compare it to standard language identification methods, employed here to detect variation in the same language.

1.1 Introduction

Variation in the same language is often regarded as a problem to categorical approaches of language, and as evidence for its probabilistic dimension (Abney 1996).

In this paper we focus on the problem of handling regional variation within a deep (categorical) natural language processing system, and present a simple way to model variation in a computational grammar using HPSG (Pollard and Sag 1994).

Support and control over variation is obviously important in these systems if they are to have practical application. On the one hand, it is desirable that such

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

systems can cope with the analysis of as many language varieties as possible, since it is less economical to write a different grammar for each language variety. On the other hand, when computational grammars are used for natural language generation, users should be able to put bounds on what is generated variation-wise. Section 1.2 presents an HPSG design to handle variation in a symbolic model.

A related issue is: if a system can be fine-tuned to a particular regional variety, what is the best way to detect whether some text that is to be processed by that system is in that variety? We present two approaches to this question.

The first approach is to use independent components that can detect the language variety being used. We hypothesize that methods developed for language identification can be used to detect varieties. Section 1.3 presents an overview and develops on two of them. The second approach is to have the computational grammar prepared for multiple language varieties, with no preprocessing necessary.

We compare the two solutions. To this end we use Portuguese, and we focus on the differences between European Portuguese (henceforth EP) and Brazilian Portuguese (BP). The methods presented are applicable to other languages.

The HPSG setup described to handle variation and the experiments were carried out with a computational HPSG currently being implemented for Portuguese. It is being developed in the LKB (Copestake 2002) and it uses MRS semantics (Copestake et al. 2001). It is part of the DELPH-IN Consortium.[1] The grammar was modest at the time of the experiments (1.6 years of development).

1.2 HPSG Implementation of Variation

In a framework like HPSG, variation can be accounted for in the feature structures manipulated by the grammar.

It is important that the grammar can work with both EP and BP because of coverage, but accepting the two will necessary increase ambiguity. The ability to control variation is important in that it is a way to control the ambiguity generated from accepting both varieties.

Control on what is generated is also desirable. In general one wants to be able to parse as much as possible (e.g. EP and BP), but at the same time be selective in generation (i.e. generate in a specific variety), so that output is tailored to the expected audience.

If a grammar accepts both EP and BP ambiguity will rise because ambiguity inevitably goes up when coverage increases. But ambiguity can be put in check by restricting the grammar to reject analyses that involve marked constructions in more than one variety. More precisely, if an input string contains an element that can only be found in variety v_1 and that same string is ambiguous in a different place but only in varieties other than v_1, this ambiguity will not give rise to multiple analyses if the grammar can be constrained to accept strings with marked elements of at most one variety.

A feature VARIANT is employed to model variation, which encodes the variety of Portuguese being used. It is appropriate for all signs and declared to be of type

[1] http://www.delph-in.net

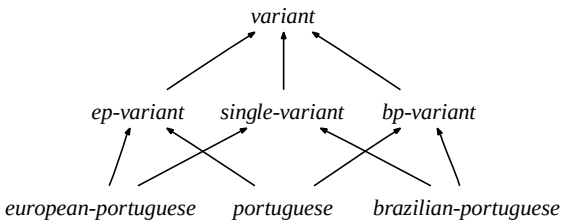


Figure 1.1: Type hierarchy under *variant*.

variant. Its possible values are presented in Figure 1.1.

This attribute is constrained to take the appropriate value in lexical items and constructions specific to one of the two main Portuguese varieties. For example, a hypothetical lexical entry for the lexical item *autocarro* (*bus*, exclusive to EP) would constrain that attribute VARIANT to have the value *ep-variant* and the corresponding BP entry for *ônibus* would constrain the same feature to bear the value *bp-variant*. The only two types that are used to mark signs are *ep-variant* and *bp-variant*. The remaining types presented in Figure 1.1 are used to perform computations or to constrain grammar behavior, as explained below.

It is not only lexical items that can have marked values in the VARIANT feature. Lexical and syntax rules can have them, too. Such constraints model marked constructions.

Feature VARIANT is structure-shared among all signs that comprise a full parse tree. This is achieved by having all lexical or syntactic rules unify their VARIANT feature with the VARIANT feature of all of their daughters.

If two signs (lexical items, syntax rules) in the same parse tree have different values for feature VARIANT (one has *ep-variant* and the other *bp-variant*), they will unify to *portuguese*, as can be seen in Figure 1.1. This type means that lexical items or constructions specific to two different varieties are used together. Furthermore, since this feature is shared among all signs, it will be visible everywhere, for instance in the root node.

It is possible to constrain feature VARIANT in the root condition of the grammar. If this feature is constrained to be of type *single-variant* (in root nodes), the grammar will accept EP and BP, but the sentences with properties of both may be blocked. As explained in the previous paragraph, feature VARIANT will have the value *portuguese* in this case, and there is no unifier for *portuguese* and *single-variant*. If this feature is constrained to be of type *european-portuguese* in the root node, the grammar will not accept any sentence with features of BP, since they will be marked to have a VARIANT of type *bp-variant*, which is incompatible with *european-portuguese*.

It is also possible to have the grammar reject EP (using type *brazilian-portuguese*) or to ignore variation completely by not constraining this feature in the start symbol.

With this mechanism, it is possible to use the grammar to detect to which variety input text belongs to. This is done by parsing that text placing no constraint on feature VARIANT of root nodes, and then reading the value of attribute VARIANT from the resulting feature structure: values *ep-variant* and *bp-variant* result from parsing text with features specific to EP or BP respectively; value *variant* indicates that no marked elements were detected and the text could be from both.

This mechanism achieves two goals:

1. Variation can be controlled. A grammar can be parameterized for language variants. Generation can be very specific by choosing values for feature VARIANT low in the type hierarchy, but good coverage variation-wise can be attained in parsing by using a more general type for the same feature. Furthermore, trade-offs between ambiguity and coverage can be explicitly controlled via intermediate types, like *single-variant*.

2. Language variants can be detected in the input.

If the input can be known to be specifically EP or BP before it is parsed, the constraints on feature VARIANT can be changed to improve efficiency. When parsing text known to be EP, there is no need to explore analyses that are markedly BP, for instance.[2]

It is thus interesting to know what other methods can do to detect varieties, and how they compare to the one just introduced, using real world data. In Section 1.3, some language identification models that can be used for this purpose are presented.

Because the two aspects of controlling variation and detecting variants are related by a single design, we assume that evaluating one indirectly evaluates the other. Therefore by investigating how good a grammar with such a mechanism can be in detecting language varieties, we can also have an idea of how well the same mechanism is used for the purpose of controlling ambiguity or specificity.

1.3 Language Detection Methods

Over the last years methods have been developed to detect the language a given text is written in. They have also been used to discriminate varieties of the same language, although less often. They can be based on words in text. Lins and Gonçalves (2004) look up words in dictionaries to discriminate among languages, and Oakes (2003) runs statistical tests on word frequencies, like the chi-square test, in order to differentiate between British and American English.

Many methods are based on frequency of byte sequences (byte n-grams) in text, because they can simultaneously detect language and character encoding (Li

[2]Currently, it is not possible to prune the parser's search space in such circumstances with the LKB, because it is only possible to constrain the root node without changing and reloading the grammar. Therefore, incompatible analyses will only be discarded when that node is built, but not before that. Efficiency could be gained if it were possible to specify constraints that all nodes in a syntactic tree must obey. The limitation is system dependent, so, in theory, efficiency can be improved in such a way.

and Momoi 2001), and can reliably classify short portions of text, since they look at such short sequences. They have been applied in web browsers (to identify character encodings as in Li and Momoi (2001)) and information retrieval systems.

We are going to focus on methods based on character n-grams. Because all information used for classification is taken from character n-grams, and they can be found in text in much larger quantities than words or phrases, sparse data problems are attenuated. Therefore, high levels of n or very small training corpora can be used. Training data can also be found in large amounts because training corpora do not need to be annotated (it is only necessary to know the language they belong to).

More importantly, methods based on character n-grams can reliably classify small portions of text. The literature on automatic language identification mentions training corpora as small as 2K producing classifiers that perform with almost perfect accuracy for test strings as little as 500 Bytes (Dunning 1994) and considering several languages. With more training data (20K-50K of text), similar quality can be achieved for smaller test strings (Prager 1999).

Many n-gram based methods have been used besides the ones we present. Sibun and Reynar (1996) and Hughes et al. (2006) present good surveys. Many can achieve perfect or nearly perfect classification with small training corpora on small texts, so we just focus on two that use approaches very well understood in language processing and information retrieval.

1.3.1 Markov Models

If one wants to know which language $L_i \in L$ generated string s, one can use Bayesian methods to calculate the probabilities $P(s|L_i)$ of string s appearing in language L_i for all $L_i \in L$, the considered language set, and decide for the language with the highest score (Dunning 1994). That is, in order to compute $P(L_i|s)$, we only compute $P(s|L_i)$. The Bayes rule allows us to cast the problem in terms of $\frac{P(s|L_i)P(L_i)}{P(s)}$, but, as is standard practice, we drop the denominator, since we are only interested in getting the highest probability score among several scores, not its exact value. The prior $P(L_i)$ is also ignored, assuming all languages are equally probable.

The way $P(s|L_i)$ is calculated is also the standard way to do it, namely assuming independence and just multiplying the probabilities of character c_i given the preceding $n-1$ characters (using n-grams), for all characters in the input string (which are estimated from n-gram counts in the training texts).

We implemented the algorithm as described in Dunning (1994) for the experiments presented in the following sections, which uses other common strategies, like prepending $n-1$ special characters to the input string to harmonize calculations, summing logs of probabilities instead of multiplying them to avoid underflow errors, and using Laplace smoothing to reserve probability mass to events not seen in training.

1.3.2 Vector Space Models

The second method using n-grams we employ in the following experiments is inspired in the vector space model of information retrieval to compare document similarity, but it uses n-gram counts instead of term frequency. It has been used for the purpose of language identification in Prager (1999).

Each language is represented by a vector built during training. Each possible n-gram corresponds to a component of that vector (e.g. if bigrams are used, the first component might represent the bigram *aa*), namely a number based on the frequency of occurrence of that that n-gram in the training corpus for that language.[3] Classification consists of creating a vector representing the input text in a similar way and choosing its nearest neighbor from the set of vectors that represent languages. The cosine of the angle between the two vectors is used as a measure of similarity. A number of well-known improvements can be used, like normalizing vectors in the training phase (make them of length = 1), so that calculating cosines amounts to calculating dot products during classification (after normalizing the vector representative of the test item).

In the literature it is also common to reduce dimensions by keeping the most frequent n-grams and discarding the rest, but we did not do this since we hypothesize that the most frequent n-grams of EP and BP will largely overlap. It has been reported that the 300 most frequent n-grams are good predictors of language, and the others are representative of the textual topic (Cavnar and Trenkle 1994).

1.4 Data and Calibration

Some preliminary studies were conducted in order to investigate the performance of the language identification methods presented above at discriminating among languages (Section 1.4.1), and to find out the impact of training corpora size when they are employed to detect language variants and what values of n are reasonable (Section 1.4.3 and Section 1.4.4). The data used in all experiments concerning variety identification are presented in Section 1.4.2.

1.4.1 Language Identification Methods at Identifying Languages

We want to check that the language identification methods we are using are in fact reliable at identifying different languages. Although the literature reports good results, we wanted to test the exact implementation we will be using in distinguishing between EP and BP.

We ran the two classifiers on three languages showing strikingly different characters and character sequences. This is a deliberately easy test to get an upper bound on what these methods can do.

[3] In information retrieval tf-idf is often used (term frequency times inverse document frequency). Here we use n-gram frequency in that language divided by the frequency of that n-gram in all languages. Both numbers are estimated from the training corpora. Note that this is a literal interpretation of inverse document frequency: it is common practice to use a value based on that instead, like its log; but Prager (1999) reports that the literal version performs better for language identification.

For this test we used the Universal Declaration of Human Rights texts.[4] The languages used were Finnish, Portuguese and Welsh. Human inspection of texts in these languages immediately reveals highly idiosyncratic character sequences.[5]

The Preamble and Articles 1–19 were used for training (8.1K of Finnish, 6.9K of Portuguese, and 6.1K of Welsh), and Articles 20–30 for testing (4.6K of Finnish, 4.7K of Portuguese, and 4.0K of Welsh). Because these methods perform better if the text they are classifying is large, several tests were conducted, splitting the test data in chunks of text 1, 5, 10 and 20 lines long.

The Bayesian method obtained perfect accuracy on all test conditions (all chunk sizes), for all values of n between 1 and 7 (inclusively). For $n = 8$ and $n = 9$ there were errors only when classifying 1 line long test items. The vector space model obtained perfect accuracy on all test conditions, for all values of n between 2 and 8 (inclusively). For $n = 1$ and $n = 9$ there were errors once again only when classifying 1 line long test items.

The average line length for the test corpora was 138 for Finnish, 141 for Portuguese and 121 for Welsh (133 overall). In the corpora we will be using in the following experiments, average line length is much lower (around 40 characters per line). Input length is obviously important for these methods. To make the results more comparable, we also evaluated these classifiers of Finnish, Portuguese and Welsh with the same test corpora, but truncated each line beyond the first 50 characters, yielding test corpora with an average line length around 38 characters (since some were smaller than that).

The results are similar, just slightly worse. The Bayesian classifier performed with less than perfect accuracy also with $n = 7$ when classifying 1 line at a time. The vector based classifier performed worse only with $n = 2$ and 1 line long test items. In all these less than perfect cases, accuracy was in the 80–90% range.

These methods thus perform very well at discriminating languages with reasonable values of n and can classify short bits of text, even with incomplete words.

1.4.2 Data

For the experiments on variety detection, we used two corpora from Portuguese and Brazilian newspaper text. They are CETEMPublico and CETENFolha. CETEMPublico contains text from the Portuguese newspaper *O Público*, and CETENFolha from the Brazilian *Folha de São Paulo*.

These corpora are minimally annotated (paragraph and sentence boundaries, *inter alia*), but are very large (CETEMPublico has 204M words and 1.2GB of text, and CETENFolha has 32M words and 183.2 MB).

[4] Available at http://www.unhchr.ch/udhr/navigate/alpha.htm.

[5] For the sake of illustration, examples (1), (2) and (3) present the first sentence of the first Article in Finnish, Portuguese and Welsh, respectively. (4) is the English version.
(1) Kaikki ihmiset syntyvät vapaina ja tasavertaisina arvoltaan ja oikeuksiltaan.
(2) Todos os seres humanos nascem livres e iguais em dignidade e em direitos.
(3) Genir pawb yn rhydd ac yn gydradd â'i gilydd mewn urddas a hawliau.
(4) All human beings are born free and equal in dignity and rights.

Some preprocessing was carried out: all XML-like tags, like the <s> and </s> tags that mark sentence boundaries, were removed. Some heuristics were also employed to remove lines that are parts of lists, like sports results tables or music charts, since they might not be representative of language: only lines ending in `.`, `!` and `?` were considered, and lines containing less than 6 words (defined as strings delimited by whitespace) were discarded. Other character sequences that were judged irrelevant for the purpose at hand were normalized: URLs were replaced by the sequence `URL`, e-mail addresses by `MAIL`, hours and dates by `HORA` and `DATA`, etc. Names at the beginning of lines indicating speaker were removed since they are frequent and the grammar that will be used cannot parse name plus sentence strings.

1.4.2.1 The 400K Line Corpus

We ordered the remaining lines by line length in terms of words and kept the smallest 200K lines from each of the two corpora. Small lines were preferred as they are more likely to receive an analysis by the grammar.

From these 200K lines of text from each corpus, we randomly chose 20K lines for testing and the remaining 180K for training. This produced a large data set, that allows one to check how good n-grams based methods can be in detecting varieties given enough data, and what values of n are necessary. Since language varieties are more similar to each other than languages, it is expected that more data or more context will be required for comparable results. In the tests below, we refer to this data set as the 400K line corpus.

We assume that the sentences from the Portuguese corpus contain text belonging to EP, and that the sentences in the Brazilian corpus represent BP text. This is a simplification, since they can contain transcriptions from speakers of the other variety. A classification is thus considered correct if the classifier can guess the newspaper the text was taken from.

1.4.2.2 The 30KB Corpus

The use of two corpora, one from EP and the other from BP, does not allow the training of n-grams based classifiers to detect sentences that are possible in both EP and BP, because only a two-way classification is present in the training data, but we want these classifiers to produce a three-way distinction. If a sentence is found in the EP corpus, one can be relatively certain that it is possible in EP, but one does not know if it is BP, too. The same is true of any sentences in the BP corpus — it can also be a sentence of EP.

To address this limitation, a native speaker of EP was asked to manually decide from sentences found in the BP corpus whether they are markedly BP or are also acceptable in EP. Conversely, a Brazilian informant detected markedly European sentences from the EP corpus.

Because this task requires manual annotation, and the methods we are employing reportedly perform well even with small training sets (when identifying

languages), we used only a small portion of text taken from these corpora.

We randomly selected 90K lines of text from each corpus and checked which ones could be parsed by the grammar. 25K lines of parsable BP and 21K of parsable EP (46K lines out of 180K, or 26%) were obtained. From these parsed lines we drew around 1800 random lines of text from each corpus, and had them annotated for whether they are possible in the other variety. Thus a three-way classification is obtained.

Perhaps not surprisingly, most of the sentences were judged to be possible in both EP and BP. 16% of the sentences in the Portuguese corpus were considered impossible in BP, and 21% of the sentences in the BP corpus were judged exclusive to it. Overall, 81% of the text was common to both varieties.

A hypothetical explanation of the asymmetry is that one of the most pervasive differences between EP and BP, clitic placement, is attenuated in writing: Brazilian text often displays word order between clitic and verb similar to EP, and different from oral BP. Therefore, European text displaying European clitic order does not look markedly European. In fact, we looked at the European sentences with clitic placement characteristic of EP that were judged possible in BP. If they were included in the markedly European sentences, 23% of the European text would be unacceptable BP, a number closer to the 21% sentences judged to be exclusively Brazilian in the Brazilian corpus.

Such information can be used to estimate prior probabilities for the Bayesian method (which, as referred in Section 1.3.1, are ignored), creating a bias for classifying text as common to all varieties of Portuguese. This was not done, because like what happens for estimating the priors of any language in a set of languages in general, the difference between the priors of EP and BP is very difficult or even impossible to obtain.

The data were split into test and training data, but only a subset of what was judged common to both varieties was kept, since that data set was much larger than the other two. 10KB of text from each class were obtained. 5KB (of each class) were reserved for training and another 5KB for test. These values are close to the ones used for language discrimination in Section 1.4.1. There are approximately 140 lines for each class. For the test corpora, we kept exactly 140 lines for each: a multiple of 20 is convenient, because we want to create chunks of 1, 5, 10 and 20 lines for testing. In the following tests, this data set is referred to as the 30KB corpus.

1.4.3 Two-way Distinction with the 400K Line Corpus

Table 1.1 summarizes the results for the n-grams trained to distinguish between EP and BP with the 400K line corpus. The average line length of the test sentences is 43 characters. Several input lengths were tried out by dividing the test data into various sets with varying size.

The accuracy of the Bayesian classifier is surprisingly high, given that we can estimate the number of sentences that cannot be attributed to a single variety to be at least 80% (see Section 1.4.2.2). We hypothesize that this is a corpus sensitiv-

Length of Test Item		1 line	5 lines	10 lines	20 lines
n = 2	Bayesian	0.84	0.99	**1**	**1**
	Vector based	0.62	0.59	0.56	0.52
n = 3	Bayesian	**0.96**	0.99	**1**	**1**
	Vector based	0.63	0.59	0.61	0.65
n = 4	Bayesian	**0.96**	**1**	**1**	**1**
	Vector based	0.63	0.73	0.79	0.87
n = 5	Bayesian	0.94	**1**	**1**	**1**
	Vector based	0.65	0.81	0.89	0.97
n = 6	Bayesian	0.92	0.99	**1**	**1**
	Vector based	0.67	0.86	0.94	0.98

Table 1.1: Precision with 360K lines of text for training, two-way classification.

ity effect. For instance, German names are more frequent in Brazil. In fact, the absolute frequency $f_{Tr}(x)$ of n-gram x in the training data for n-grams *sch*/*Sch*, *ung*, *W* and *en*␣[6] is $f_{Tr}(sch \vee Sch) = 311$, $f_{Tr}(ung) = 194$, $f_{Tr}(W) = 1122$ and $f_{Tr}(en␣) = 529$ in Brazilian text and $f_{Tr}(sch \vee Sch) = 205$, $f_{Tr}(ung) = 98$, $f_{Tr}(W) = 680$ and $f_{Tr}(en␣) = 305$ in Portuguese text. This might also explain the lower performance of the vector space model, where infrequent n-grams have a lower impact on the result since the individual values derived from the n-grams are summed together rather than multiplied.

The amount of training data is very large because these methods look at characters. There are 15.5M of them in the training sets. The fact that relatively high values of n (4 and 5 for 5 lines of input) are necessary to achieve perfect accuracy on small inputs (and perfection is never found with 1 line long test items) suggests that variety discrimination is much harder than language identification.

1.4.4 Two-way Distinction with the 30KB Corpus

The same experiment was conducted, using only the EP and BP data (not the sentences judged to be common to both) of the 30KB corpus (only 20KB of it).

Although the size of training data is much smaller than in the test reported in Section 1.4.3, the two classes are expected to be farther apart since sentences judged to be common to the two varieties were not included.

The results are in Table 1.2. The Bayesian classifier is very good with bigrams, but because of the small training data, it is heavily biased at classifying everything as EP. The vector space model cannot achieve as good performance with bigrams, but is less affected by sparseness of training data.

[6] ␣ denotes a space.

Length of Test Item		1 line	5 lines	10 lines	20 lines
$n = 2$	Bayesian	**0.86**	**0.98**	**0.96**	**1**
	Vector based	0.61	0.75	0.86	0.85
$n = 3$	Bayesian	0.82	0.73	0.64	0.5
	Vector based	0.64	0.61	0.71	0.79
$n = 4$	Bayesian	0.68	0.55	0.5	0.5
	Vector based	0.64	0.71	0.79	0.93

Table 1.2: Precision with 10K lines of text for training, two-way classification.

1.4.5 Differences between EP and BP

We proceeded to an analysis of the training data resulting from the manual classification described in Section 1.4.2.2 (the 30KB corpus). A brief typology of the markedly Brazilian elements found in the BP training corpus is presented. We also present the relative frequency of these phenomena based on the same data.

1. Mere orthographic differences (24%)
   e.g. *ação* vs. *acção* (*action*)
2. Phonetic variants reflected in orthography (9.3%)
   e.g. *irônico* vs. *irónico* (*ironic*)
3. Lexical differences (26.9% of differences)
   (a) Different form, same meaning (22.5%)
   e.g. *time* vs. *equipa* (*team*)
   (b) Same form, different meaning (4.4%)
   e.g. *policial* (*policeman/criminal novel*)
4. Syntactic differences (39.7%)
   (a) Possessives without articles (12.2%)
   (b) In subcategorization frames (9.8%)
   (c) Clitic placement (6.4%)
   (d) Singular bare NPs (5.4%)
   (e) In subcat and word sense (1.9%)
   (f) Universal *todo* occurring with article (0.9%)
   (g) Contractions of preposition and article (0.9%)
   (h) Questions without subject-verb inversion (0.9%)
   (i) Postverbal negation (0.5%)
   (j) other (0.5%)

One third of the differences found would be avoided if the orthographies were unified (items (2) and (1)).

Some differences cannot be detected by n-gram based methods or the grammar. This is the case of item (3b), which would require word sense disambiguation. When word sense differences are accompanied by different syntax, they can be detected by the grammar (item (4e)) in limited circumstances (in that example, only if the complement is expressed).

Differences that are reflected in spelling can be modeled by the grammar via multiple lexical entries, with constraints on feature VARIANT reflecting the variety in which the lexical with that spelling item is used.[7]

Interestingly, 40% of the differences are syntactic. These cases are expected to be difficult to detect with n-gram based approaches, but not by a grammar.

Note that on average each sentence contained 1.46 marked elements. Spelling differences (items (2) and (1)), which account for 33.3% of all differences appear in 47.9% of them (in the BP training corpus). N-grams models can detect them.[8]

In the Portuguese grammar we use for the experiments, only clitic word order (item (4c)) and co-occurrence of prenominal possessives and determiners (item (4a)) are marked with respect to the VARIANT feature. The main limitation is grammar immaturity, in that several differences involving phenomena that are not implemented yet cannot be taken into account. These two phenomena do account for 18.6% of the differences found.

We expanded the grammar with many lexical items markedly EP or markedly BP. These were taken from the Portuguese Wiktionary,[9] where this information is available. We did not include all of the ones there, since some were judged infrequent and manual expansion of a lexicon for a deep grammar is time consuming. At the end, around 740 lexical items were added. Variety specific lexical items found in the training corpora (80 more) were also incorporated in the lexicon.

1.5 Results

We report on the evaluation of the n-gram based methods presented in Section 1.3 and the grammar-based mechanism to handle variation described in Section 1.2, tested with the 30KB corpus (Section 1.4.2.2).

When the grammar produced multiple analyses for a sentence, we only considered that sentence to be classified as EP if all the parses produced VARIANT with type *ep-variant*, and similarly for BP. In all other cases the sentence would be considered common to both.

The grammar can only look at one line at a time, but several input sizes are tested. In order to make the grammar results comparable, this is done also for the

[7] In some cases a different solution would be preferable. When the difference is systematic (e.g. the EP sequence *ón* always corresponds to a BP sequence *ôn*, with an example in item (2)), it would be best to have a lexical rule that affects only spelling and the VARIANT feature producing one variant from the other. This is not implemented, because string manipulation is limited in the LKB.

[8] Even when these differences are not absolute, they are often strongly unbalanced. For instance, the bigram *ct* appears 22 times in the Portuguese training corpus and only once in the Brazilian one.

[9] http://pt.wiktionary.org

Length of Test Item		1 line	5 lines	10 lines	20 lines
Grammar	Precision	0.57	**0.78**	0.72	0.64
	Recall	0.57	**0.72**	0.62	0.43
	$F_{\alpha=1}$	0.57	**0.75**	0.67	0.51
n = 2	Bayesian	**0.59**	0.67	**0.76**	**0.76**
	Vector based	0.43	0.52	0.55	0.57
n = 3	Bayesian	0.55	0.52	0.45	0.33
	Vector based	0.47	0.48	0.67	**0.76**
n = 4	Bayesian	0.48	0.39	0.33	0.33
	Vector based	0.41	0.5	0.71	0.67

Table 1.3: Evaluation of variety identification, three-way classification. With the n-grams based method, precision, recall and the F-measure are identical under the same conditions.

grammar. In this case, the result for chunks of more than one line is the unification of the values for each line. If the unification result is *portuguese* (see the hierarchy in Section 1.2), signaling inconsistency, the grammar does not decide, affecting recall but not precision. For this reason, precision, recall, and the F-measure can be different and are all reported. With the n-grams based models, they are always identical. The results for the three-way classification are in Table 1.3.

Error analysis shows that the BP sentences classified as EP contain clitics following the EP syntax, and misspellings conforming to the EP orthography.[10] Most of the sentences common to EP and BP that were classified as EP also present clitics with this syntax. A large proportion of the errors consisted in classifying as common to all varieties of Portuguese sentences that were in fact marked. Inspection of these sentences reveals many marked lexical items.[11] It is thus a problem of lexical coverage.

The Bayesian method works well with small values of n, but it tends to classify everything as EP, producing correct classifications for only one third of the test items. The vector space model is more affected by input length.

1.6 Discussion and Conclusions

Before getting into the analysis of the quantitative results obtained above, some remarks on the two approaches, with the grammar and with the stochastic techniques, follow from the very nature of these methods.

Bayesian and vector-similarity methods are expected to be easier to scale up with respect to the number of varieties considered given that the size of the type hierarchy under *variant* is exponential on the number of language varieties if all

[10] In Brazil a diaeresis is used on *u* when it follows *q*, precedes *e* or *i* and is pronounced. In the Portuguese orthography it is no longer used. The errors were due to the spellings *aguentar* (*to bear*) and *tranquilo* (*calm*), instead of *agüentar* and *tranqüilo*.

[11] Note that, in order to increase grammar coverage, we used a POS tagger to get information about unknown words. Obviously, feature VARIANT was left underspecified in these items.

variety combinations are taken into account.[12]

In turn, provided the symbolic method is supported by a more matured grammar than the one we could use in the present experiments, with a large enough lexicon, stochastic methods are expected to show more dependency on the text domain they are applied to than the grammar, and it is likely that their performance tends to degrade more severely when applied over texts from domains which they were not trained with.

Focusing on the results obtained with the grammar, the fact that the best score results from setting the input with 5 lines/sentences is understandable at the light of the following considerations: on the one hand, taken individually, there is a certain chance that each sentence ends up not being specified with respect to any language variant at stake; on the other hand, when they are bundled together, there happens the incremental effect that the resolution obtained at one or several of them in each bunch unifies with the underspecified values of the remaining ones that did not get resolved; however, when they are bundled into a too large bunch ($\geq$ 10 lines/sentences) chances also increase that different sentences get different specifications, which induces incorrect or even non resolution for the whole bunch, thus canceling the beneficial effect of the sentences being bundled together.

By the same token, it is also worth noting that with larger bundles, the performance of classifiers based on the grammar is thus expected to degrade more than the performance of classifiers based on n-grams.[13]

Note however that this may not be a shortcoming for the grammar-based methods in every application scenario. For instance, when the input text is a dialog, such input may have to be entered in small chunks (a chunk per turn) if one wants to contemplate conversations between speakers of different varieties.

Turning now to the evaluation results obtained above, both the grammar and the stochastic approaches displayed similar results. In both cases the best score is around $F = 0.75$.

For both approaches, our experiments were limited in several respects and there is plenty of room for improvement. The n-grams methods can be enhanced by using more training data, since only 15KB were used. With the grammar, lexical coverage can be augmented, and more marked constructions can be added — the syntactic differences considered cover half of the occurrences of all syntactic differences found in the BP training data (Section 1.4.5).

In spite of the limitations of these first experiments, results are encouraging. The design we presented to account for variation can be adapted to other feature-type formalisms, and the experimental setup used to compare performance in face of language varieties, which takes into account the fact that they largely overlap, is new and extensible to other languages as well.

[12] This may be necessary. For instance, *bué* (*very*, *much*) is a word in European and Angolan Portuguese, but not in Brazilian Portuguese; *moleque* (*boy*) is a word in Angolan and Brazilian Portuguese, but not in EP, etc.

[13] Recall for common Portuguese is 0.89 in the 1 line test, and 0.14 in the 20 lines case. Overall, 68% of the test items were classified as common in the 1 line test, but only 5% in the 20 lines test.

References

Abney, S.(1996), Statistical methods and linguistics, *in* J. Klavans and P. Resnik (eds), *The Balancing Act*, The MIT Press, Cambridge, MA.

Cavnar, W. B. and Trenkle, J. M.(1994), N-gram-based text categorization, *Proceedings of the 1994 Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV USA.

Copestake, A.(2002), *Implementing typed feature structure grammars*, CSLI Publications, Stanford, California.

Copestake, A., Flickinger, D., Pollard, C. and Sag, I. A.(2001), Minimal Recursion Semantics: An introduction, *Language and Computation*.

Dunning, T.(1994), Statistical identification of language, *Technical Report MCCS-94-273*, Computing Research Lab (CRL), New Mexico State University.

Hughes, B., Baldwin, T., Bird, S., Nicholson, J. and MacKinlay, A.(2006), Reconsidering language identification for written language resources, *Proceedings of LREC2006*, Genoa, Italy.

Li, S. and Momoi, K.(2001), A composite approach to language/encoding detection, *Proceedings of the Nineteenth International Unicode Conference*.

Lins, R. D. and Gonçalves, P.(2004), Automatic language identification of written texts, *Proceedings of the 2004 ACM Symposium on Applied Computing*.

Oakes, M. P.(2003), Text categorization: Automatic discrimination between US and UK English using the chi-square test and high ratio pairs, *Research in Language*.

Pollard, C. and Sag, I.(1994), *Head-driven phrase structure grammar*, Chicago University Press and CSLI Publications.

Prager, J. M.(1999), Linguini: Language identification for multilingual documents, *Journal of Management Information Systems*.

Sibun, P. and Reynar, J. C.(1996), Language identification: Examining the issues, *5th Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, U.S.A.

2

A Vector-based Approach to Dialectometry

Erhard Hinrichs and Thomas Zastrow
University of Tübingen

Abstract

A novel unsupervised learning approach to computational dialectometry is presented which uses hard clustering. The approach relies on vector analysis over two-dimensional arrays of word lists collected for different geographical sites. The paper presents the underlying theory and applies the approach to a Bulgarian data set. The results of these experiments demonstrate the viability of the approach.

2.1 Computational Dialectometry

The study of language variation and of language change has a long and venerable tradition in linguistics. Traditional dialectology deals with the identification of dialect boundaries on the basis of historical evidence and on the basis of bundles of characteristic isoglosses. Relevant historical evidence includes information about lanugage contact, migration and settlement patterns, as well as processes of urbanisation. Isoglosses refer to dialect boundaries determined by individual linguistic features (such as word pronunciations, lexical choice or syntactic constructions). In contrast, computational dialectometry clusters phonetic or lexical data in the form of word lists into geographical dialect-regions by means of quantitatively defined distance measures (Göbl 1982, Kessler 1995, Nerbonne 2006, Nerbonne and

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

Hinrichs 2006, Prokić 2006).

Currently the method of choice for measuring the distance between two words, either in terms of their graphemic representation (in the case of lexical data) or in terms of their phonetic representation (in the case of pronunciation data), relies on the notions of alignment (Kondrak 2000) and of edit-distance (Heeringa 2004), particularly in the form of Levenshtein-Distance. The distance between lists of words is measured by an aggregate method that provides the summation of the distances in the word list.

Two disadvantages are implicit in these approaches:

- Aggregate methods consider in every pass just two data-records, not the entire data set. A comparison of the whole data set in a single step is not possible.
- It is only possible to compare pairs of individual words. For example, it is possible to compare two different pronunciations of the word *apple*, but it is not possible to track the occurrences of individual segments, e.g. the vowel *a* in different words, e.g. in *apple* and *banana*.

In this paper a new approach to computational dialectometry is proposed that is based on vector analysis and that avoids the above disadvantages of the aggregate-method. The approach is inspired by the Neogrammarian notion of regular sound correspondences. This notion has played a major role in the study of language change. Here it is applied to the study of language variation.

2.2 The Data

2.2.1 General format

The data takes the form of word lists, one such list per site. A site is a geographicaly defined point like a village or a town. Other properties such as size, geographical properties, more rural or more urban, are ignored at present.

For every site, the same words are collected and transcribed into X-Sampa (Wells n.d.), which is an electronic readable form of the IPA, the International Phonetic Alphabet (IPA 2003). The X-Sampa codes are the smallest units in the data-sets.

This data format allows investigations in two directions:

- Horizontal: In this direction all occurrences of a given element are traced in a single word from the word list across all sites. We will henceforth refer to such a horizontal trace as *single-word-all-sites* (SWAS-trace).
- Vertical: In this direction all occurrences of a given element are traced across the entire word list for a single site. We will henceforth refer to such a vertical trace as *single-site-all-words* (SSAW).

In the horizontal dimension, comparisons of a given element across different pronunciations of the same lexical item can detect regularities and irregularities

	Aldomirovci	Asparuhovo	...	Zheravna
агне (lamb)	"jAgne	"Agni	...	"Agni
аз (I)	"jA	"As	...	"As
бели (white-plural)	"beli	"beli	...	"beli
берат (pick up - 3rd plural)	"beru	bi"r7t	...	bi"r7t
...	...	...	...	...
ям (eat, 1st singular)	e"dem	"jAm	...	"jAm

Figure 2.1: general data-structure

of sound correspondences in the set of pronunciations. In the vertical dimension, comparisons of a given element across the word lists of different sites can reveal phonological and/or morphological proccesses such as insertion, deletion, and metathesis which are commonly found in language variation.

2.3 The Bulgarian Data-Set

In cooperation with the Bulgarian Academy of Science and the University of Sofia, a phonetic data-set of the Bulgarian language with 200 sites and 143 words[1] common to all sites is been collected (Osenova and Simov 2005). These 200 sites are spread across the whole territory of Bulgaria. At the moment, 121 sites are available in electronic form. XML is used as a container for the data. This data set forms the basis for all vector-based experiments reported in this paper.

(Zhobov 2006) provides detailed information about the selection of words that have been chosen for the data set and about the sources that have been consulted for their pronunciation.

2.4 The Vector-based Approach

2.4.1 Background: Vector Analysis

Vector analysis is a subarea of geometry. It deals with arrays (vectors) in a two- or higher dimensional space. In these spaces, vectors are defined by two points, each identified by one coordinate for each dimension. The arrays in our particular dialectometry application are always two-dimensional (one dimension for the canonical order of words in the word list and one for the order of elements within the individual word). Figure 2.2 gives an example of a vector $\vec{v_1}$ in two-dimensional space with the starting point (2,2) and the end point (4,4):

The length of a vector can be calculated on the basis of the Pythagorean theorem:

$$|\vec{v_1}| = \sqrt{\triangle x^2 + \triangle y^2} \tag{2.1}$$

[1]Some of the sites contain more words, but these 143 words are included in every site.

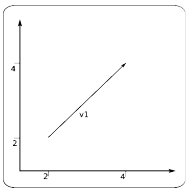


Figure 2.2: a two-dimensional vector

where $\triangle x$ and $\triangle y$ are the relative position-changes of the vector on the X and the Y axis.

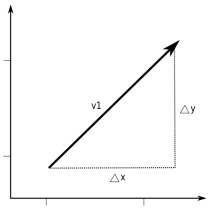


Figure 2.3: calculating the length of a vector

To compute the angle between two 2-dimensional vectors:

$$cos(\alpha) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \tag{2.2}$$

2.4.2 The Algorithm

In this method one element of interest is selected. This can be a single segment, a bigram or even longer sequences of segments. By the use of vectors, the element in focus is traced either horizontally (SWAS) or vertically (SSAW) through the entire data set. Each occurrence of the focus item is represented by a single vector. Combining these vectors into a chain of vectors, the relative position changes of the relevant element are recorded. The pseudocode for constructing such a vector chain for an *SWAS* trace or an *SSAW* trace is shown in figure 2.4.

On the X-axis of the coordinate system the units of measurement are the positions of X-Sampa codes in individual words. On the Y-axis, the words are the unit of measurement. By assumption, a shift on the X-axis of one X-Sampa to the left or to the right has the value 1. On the Y-axis going down one line to the next word, without shift on the X-axis, has the value of 1.

A vector chain constitutes a unique fingerprint of the occurrence of the element in focus either in a single word across all sites in the horizontal dimension or in

```
delta[X] = 0;
delta[Y] = 0;

for i=1 to number of occurrences of element A

    delta[X] = X(A[i]) - delta[X];
    delta[Y] = Y(A[i]) - delta[Y];
    addToVectorChain(<delta[X], delta[Y]>);
```

Figure 2.4: pseudocode for constructing a vector chain for a single focused element

all words of the word list for a single site in the vertical dimension. Moreover, in each dimension such fingerprints can be compared across sites or across words.

In the following example (Figure 2.5), a hypothetical element A is followed through a data record. The origin and starting point of the first vector is set to the first element of the data record.

Starting in the upper lefthand corner (0,0), the first appearance of "A" can be achieved by the vector $\vec{v} = (3, 0)$. The first coordinate represents the movement on the X- and the second one the movement on the Y-axis: this means, that they are showing the *relative* movement of a vector from the actual element to the next one, not the absolute position in the coordinate-system. From here, a second vector is drawn down to the second appearance of "A" (-1, +1), and so on:

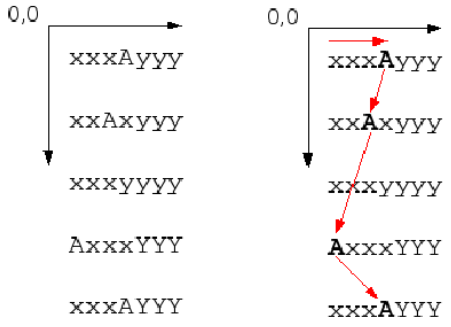


Figure 2.5: artificial example for tracing an element

Figure 2.6 shows an excerpt from the Bulgarian data set. From left to right: The first 13 words of the site Rakovica, located in western Bulgaria. In the middle, the vector chain for the vowel "e" is drawn. On the righthand side, the complete "e"-vector for the 143 words of site Rakovica is shown:

Figure 2.6: from left to right: partial word list, partial vector chain, complete vector chain

2.4.3 The Length of a Vector Chain

In the previous section, we have shown how vector chains can be created. Graphically such vector chains can be rendered as shown in Figure 2.6. However, in order to be able to compare vector chains with one another, a quantitative measure is needed. Such a measure can be obtained on the basis of the length of a vector chain[2]. This length can be calculated by adding together the lengths of the individual vectors contained in the vector chain.

$$|\vec{v_c}| = \sum_{i=1}^{n} \sqrt{\triangle x_{v_i}^2 + \triangle y_{v_i}^2} \tag{2.3}$$

where *n* is the number of single vectors in the vector chain.

For illustration, Figure 2.7 shows some typical vector chains and their lengths:

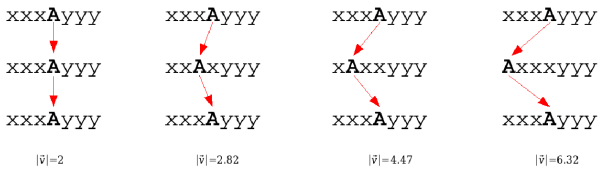


Figure 2.7: some typical vector lengths

[2]Another possibility would be to sum up the absolute movement of the element to the right and to the left. This *fluctuation* has one disadvantage: it cannot handle words which has more than one element correctly.

When calculating the length of vector chains, two questions arise: first, how to treat words with zero occurrences of the element and second, what to do when a word contains more than one occurrence of the same element. If a word has no occurrence of the element in focus, the vector chain will pass through this word and will extend to the next word in the word list that contains the element in focus. In consideration of the Pythagoraean Theorem, the resulting length of such a vector chain differs from a vector chain where each word contains exactly one occurrence of the focused element. This is illustrated in figure 2.8.

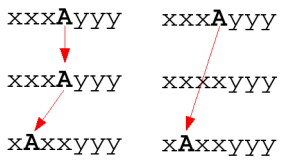


Figure 2.8: vectors when in a word the element doesn't occur

If a word has more than one element "A", additional vectors are drawn (Figure 2.9).

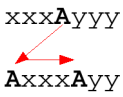


Figure 2.9: vectors when an element occurs more than one times in a word

Depending on the direction of analysis (horizontal or vertical), the length of a vector chain can be interpreted in two ways:

- In the horizontal direction: a higher value means that the specific element has more fluctuation than with a lower value. Elements with a high value are of particular interest since they carry a high degree of information about the linguistic distance across sites.
- In the vertical direction: the vector chain provides a site-specific, individual "fingerprint" of that element. The values of the individual vector chains for each site can then be clustered as described in more detail in section 2.5.

Some prominent values for the vector length are:

- If the element doesn't appear in the complete data-set, the length of the vector chain is zero.
- If the element always appears at the same position, the vector chain's length is identical to the number of words in the chain: there is just movement on the Y-, but none on the X-axis.

- The maximum length of a vector chain depends on the length of the words and their order

2.5 Vector-based Analysis: Selection and Clustering of Elements

As described in section 2.1, computational dialectometry deals with geographically defined dialect regions. Considering this goal, the examination in the vertical direction, which yields site-specific fingerprints is of central concern. This constitutes one more contrast to aggregate methods where pairwise comparison of individual words (in the horizontal dimension) provide the most important dataset.

In the vector-based analysis, the horizontal dimension can be used to determine which elements carry the highest degree of information about the linguistic distance between individual words. The elements with the highest information content can be selected to create particularly content-rich fingerprints of individual sites.

Such fingerprints can then be used to cluster the sites. The clustering is done by a bottom-up, hard clustering algorithm. Hard clustering is used so that each site can appear as a member in exact one cluster. Clustering proceeds in an iterative fashion. At the beginning, every site is its own cluster. In subsequent iterations clusters are merged until a fixed number of clusters has been reached. The target number of clusters is set in advance and depends on the desired granularity of geographic distribution.

2.6 Experiments with the Bulgarian Data-Set

This section reports on the application of the vector-based analysis, whose underlying theoretical assumptions have been presented in the previous sections, to the Bulgarian data set introduced in section 2.3. These experiments follow the strategy outlined in section 2.5. In a first step, an SWAS trace is performed for all single element X-Sampa codes contained in the entire data set. In a second step, the most content-rich elements are identified. In a third step, a SSAW trace is performed, which generates site-specific fingerprints for each of these most content-rich elements. In a fourth step, the lengths of the vector chains for each of these fingerprints is computed as described in section 2.4.3. Finally, these characteristic lengths are used in the hard clustering algorithm that was described in the previous section.

2.6.1 Finding content-rich Elements

Figure 2.10 shows the results of the first analysis steps. It displays the 10 most content-rich segments rendered in their respective X-Sampa codes.

Notice that, most of these elements are vowels or semi vowels (palatalized j). This quantitative finding corroborates the often-cited observation by traditional dialectologists that vowels tend to exhibit the highest degree of dialect variation.

X-Sampa-Code	Length of Vector chain
e	40015.1759910523
stress	35731.207131129
7 *(close-mid back, unrounded)*	35653.6778159966
A	35432.7572223606
i	34438.756791175
u	34120.3965759371
n	33581.1330654058
s	33038.0473845845
o	32878.0780176776
_j *(palatalized)*	32317.4612226377

Figure 2.10: the 10 most content-rich segments in the Bulgarian data set

The fact that the vector-based analysis is able to induce this observation by purely automatic means attests the viability of this method.

There is a second finding contained in Figure 2.10 that directly conforms to observations found in the traditional literature on Bulgarian dialect variation. It is the observation that different stress placements play a prominent role in the identification of dialect regions. Once again, the vector-based analysis induced this finding by purely quantitative means since the X-Sampa code for stress is identified as the second most content-rich element.

2.6.2 Creating Vector Chains

With the use of the elements in Figure 2.10, vector chains can be build for every site. Figure 2.11 shows 6 of these fingerprints, using the X-Sampa code “e”. Three of these sites are located in the eastern part and three in the western part of Bulgaria. Figure 2.12 shows their exact locations. The graphical rendered fingerprints of the 6 sites shows that individual fingerprints are an indicator for the sites’ geographical position.

2.6.3 Clustering the Sites

For every of the above described fingerprints the length of the vector chain can be computed. This results in a single value for every site, representing the variation of the focused element.

Using the above described clustering algorithm on these values, a distinction between the eastern part and the western part of Bulgaria can be seen for the entire data set in Figure 2.13. This east/west split once again conforms to the claim found in the traditional literature that the major division among Bulgarian dialects follows this orientation.

This distinction between the east and the west of Bulgaria can be seen in nearly every element. In general, the vowels are producing better results than the conso-

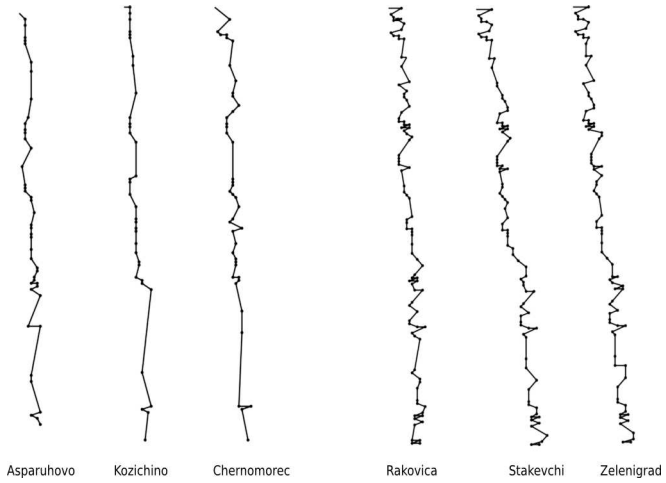


Figure 2.11: fingerprints of six sites, three in the east and three in the west of Bulgaria

Figure 2.12: the locations of the six sites

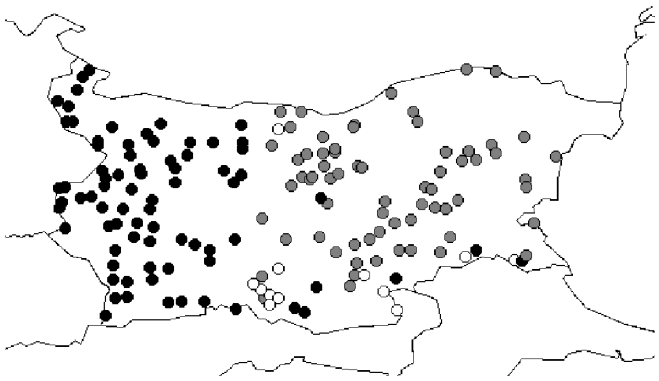


Figure 2.13: east-west distinction of Bulgaria, using the X-Sampa code “e”

nants since the variability of vowels is by comparison much larger than that of consonants.

2.7 Conclusion

A novel unsupervised learning approach to computational dialectometry is presented which uses hard clustering. The approach relies on vector analysis over two-dimensional arrays of word lists collected for different geographical sites. The paper presents the underlying theory and applies the approach to a Bulgarian data set. The results of these experiments demonstrate the viability of the approach since it is able to reproduce by purely quantitative means the major findings that have been obtained by traditional methods of dialectology for Bulgarian language variation.

In future research we plan to conduct further experiments with the full Bulgarian data set once it has become available. A second future field of experimentation concerns the length of the elements traced in the horizontal or vertical dimension. In the experiment described in section 2.6 we only investigated unigrams. Further experiments with bigrams and trigrams need to be conducted.

A third direction for further experimentation concerns the order of words in the word list. Currently, the words are ordered alphabetically. An anonymous reviewer has raised the issue whether the results depend on the order of the words and has suggested to compare the vector chains for different random permutations in the word list.

Finally, in the current experiments, we only used a single hard clustering approach. The investigation of different variants of hard clustering could well be another area where the current results may be improved.

2.8 Acknowledgement

This research was conducted by a collaborative project entitled *Measuring linguistic unity and diversity in Europe* with the following project partners: Bulgarian Academy of Science, Sofia, Bulgaria; Rijksuniversiteit Groningen, Alfa-informatica; Eberhard Karls University Tübingen, Seminar für Sprachwissenschaft. We should like to acknowledge the financial support given by the Volkswagen-Stiftung for this project[3].

We are much indebted to our colleagues in the project, in particular to Georgi Kolev, John Nerbonne, Petya Osenova, Petar Shishkov, Kiril Simov, and Vladimir Zhobov, for extended discussions on scientific matters related to this paper.

References

Göbl, H.(1982), *Dialektometrie. Prinzipien und Methoden des Einsatzes der Numerischen Taxonomie im Bereich der Dialektgeographie*, Denkschriften der Österreichischen Akademie der Wissenschaften, phil.-hist. Klasse, Österreichischen Akademie der Wissenschaften, Vienna.

Heeringa, W.(2004), *Measuring Dialect Pronunciation Differences using Levenshtein Distance*, PhD thesis, University of Groningen, Groningen.

IPA(2003), *Handbook of the International Phonetic Association*, Cambridge University Press.

Kessler, B.(1995), Computational Dialectology in Irish Gaelic, *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL-1995)*, Association for Computational Linguistics.

Kondrak, G.(2000), A new algorithm for the alignment of phonetic sequences, *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, Seattle, pp. 288–295.

Nerbonne, J.(2006), Identifying linguistic structure in aggregate comparison, *Literary and Linguistic Computing* **21**(4), 463–476.

Nerbonne, J. and Hinrichs, E. (eds)(2006), *Linguistic Distances*, Proceedings of the ACL-COLING-2006 Workshop, Association for Computational Linguistics, Sydney.

Osenova, P. and Simov, K.(2005), An infrastructure for storing and processing dialect data, Unpublished manuscript, Bulgarian Academy of Sciences. Available at: www.sfs.uni-tuebingen.de/dialectometry/documents.shtml.

Prokić, J.(2006), *Identifying Linguistic Structure in a Quantitative Analysis of Bulgarian Dialect Pronunciation*, Master's thesis, University of Tübingen.

Wells, J.(n.d.), Computer-coding the IPA: A Proposed Extension of SAMPA, http://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm.

[3]See `http://www.sfs.uni-tuebingen.de/dialectometry/` for further information

Zhobov, V.(2006), Description of the Sources for the Pronunciation Data, Unpublished manuscript. Department of Slavic Philologies, University of Sofia.

3

Comparing Improved Language Models for Sentence Retrieval in Question Answering

Andreas Merkel and Dietrich Klakow
Saarland University

Abstract

A retrieval system is a very important part in a question answering framework. It reduces the number of documents to be considered for finding an answer. For further refinement, the documents are split up into smaller chunks to deal with topic variability in larger documents. In our case, we divided the documents into single sentences. Then a language model based approach was used to re-rank the sentence collection.

For this purpose, we developed a new language model toolkit. It implements all standard language modeling techniques and is more flexible than other tools in terms of backing-off strategies, model combinations and design of the retrieval vocabulary. With the aid of this toolkit we conducted re-ranking experiments with standard language model based smoothing methods. On top of these algorithms we developed some new, improved models including dynamic stop word reduction and stemming. We also experimented with query expansion depending on the type of a query. On a TREC corpus, we demonstrate that our proposed approaches provide a performance superior to the standard methods. In terms of

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

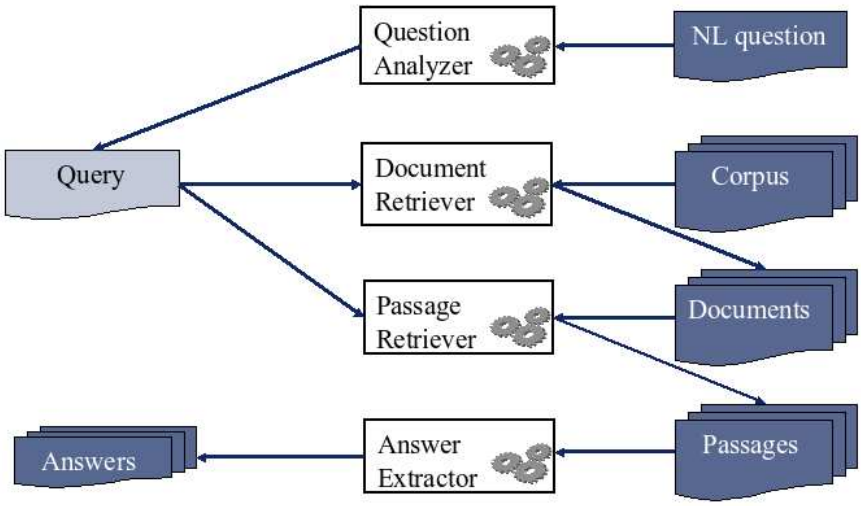


Figure 3.1: A general architecture for question answering systems

Mean Reciprocal Rank (MRR) we can prove a performance gain from 0.31 to 0.39.

3.1 Introduction

The major goal of a question answering (QA) system is to provide an accurate answer to a user question. Compared to a standard document retrieval framework, which just returns relevant documents to a query, a QA system has to respond with an adequate answer to a natural language question. Thus, the process of retrieving documents is just a part of a complex sequence. In order to provide the user with an answer, possible candidates have to be extracted from the documents. To simplify this procedure, the text is segmented into smaller passages and a further retrieval step is done. This process is called sentence retrieval, if the passage contains just one sentence.

In this paper, we describe an experimental setup for comparing different language models to improve sentence retrieval within a question answering context[1]. Figure 3.1 shows the general construction for a question answering system. It starts with the analysis of a natural language (NL) question (upper part). Generally, in this *Question Analyzer*, the expected answer type is determined, but it is also possible to make some other deep analyses like part of speech (POS) tagging or named entity recognition. The result is a processed query, which can be used for the following retrieval steps.

The next step is the document retrieval (*Document Retriever*). In a QA system, the retrieval framework is a very crucial part. It is used to decrease the number of

[1]This work was partially funded by the BMBF project SmartWeb under contract number 01 IMD01 M.

documents in a potential large corpus. This is done in order to reduce the search space in which a correct answer has to be found. It is necessary to reduce the search space because the following components may use long-lasting deep analysis algorithms which strongly depend on the size of the processed corpus. Therefore, it is important to process just the documents which seem relevant to a query to get answers within an appropriate period of time.

But within such a limited collection, there might still be large documents. Or within single documents some topic changes occur. If this is the case, again, the following components have to analyse more text than is necessary in order to find the correct answer. To overcome this problem, it is essential to further reduce the size of the collection. This can be done by splitting up the text segments into smaller chunks of passages[2]. After dividing the documents, a second retrieval step is necessary in order to re-rank the new passage collection (*Passage Retriever*) using the pre-processed query. By doing so, the corpus size and thus the search space is reduced again. One can say that the major goal of information retrieval in question answering is to achieve a high precision at a small collection of text segments (Corrada-Emmanuel et al. 2003).

In a final step, the passage collection is processed by the *Answer Extractor*. Here, the single passages are analyzed by computing the part of speech, the named entities and other linguistic features. Finally, the most probable answers are selected and returned by the system.

For our experimental setup we did not use the complete general architecture of a question answering system but just the upper part of fig. 3.1. So we skipped the extraction of the most relevant answers.

In the query construction, we used a language model driven method so as to find the expected answer type (Merkel and Klakow 2007) and some simple techniques to optimize the question for the following retrieval steps. The document retrieval was also done, using a language modeling approach.

Nevertheless, in this paper, we used a special case of passage retrieval where we directly split the documents into single sentences. In a next step, a language model (LM) based approach with unigram distributions was used to re-rank the text chunks. For these purposes, we developed a new language model toolkit. It implements all standard language modeling techniques, like linear interpolation and backing-off models. Its advantage is that it is more flexible than other tools in terms of model combinations, design of the retrieval vocabulary and the smoothing strategies. By means of this toolkit we conducted re-ranking experiments with standard language model based smoothing methods like Jelinek-Mercer linear interpolation, Bayesian smoothing with Dirichlet priors and absolute discounting as well as some new, improved models. We focused on investigating refinements which are easy to implement such as ignoring query words, dynamic stopword lists and stemming. We also experimented by modeling the expected answer type of a query into the LM approach.

To make our results comparable to current literature, we evaluated our algo-

[2]Popular methods to make passages and research into the effects on retrieval can be found in Clarke et al. (2000) and Tellex et al. (2003).

rithms on a news texts corpus from the Text REtrieval Conference (TREC) – the Aquaint corpus. Here, we demonstrate that our proposed algorithms outperform the standard methods in terms of mean reciprocal rank (MRR) by 25%. We can also show that we need to return fewer sentences to achieve equal or even better accuracy. So it is possible to say that we attained our goal, namely to reduce the search space for the following components in a QA framework.

The rest of paper is organized as follows: The next section presents some related work. Sect. 3.3 shows the language model based smoothing methods we used for our experiments. Sect. 3.4 presents the used datasets as well as the experiments we performed in order to achieve optimal results. Sect. 3.5 concludes the results.

3.2 Related Work

In the area of passage retrieval for question answering, one comes across a lot of secondary literature. For example, Clarke et al. (2000) introduces a passage retrieval system for the TREC *Question Answering track*. They use a question pre-processing, a passage retrieval and a passage post-processing step to select the top five text sections out of a set of documents. In the pre-processing step, the question is parsed and "selection rules" (patterns) are defined. Each text block for the passage retrieval algorithm can start and end with any query term. The score of such a passage is calculated by the text size and the number of occurring query terms. Then a new passage with a required length around the center point of the original passage is produced. Finally, the patterns are used to post-process the passage retrieval results.

Tellex et al. (2003) provides an overview of various state-of-the-art passage retrieval systems for question answering. They built a framework for the use of different document retrieval and passage making systems to compare the results. The text selection methods are re-implementations of famous TREC systems like MITRE, bm25, IBM and MultiText. The most important findings of their work is that boolean querying performs well for question answering, that the choice of the document retriever is very important and, that the best algorithms use density based scoring.

There is also some literature in the field of language model based passage retrieval for QA. In Zhang and Lee (2004) LM based question classification and a LM based passage retrieval approach is shown. To optimize their text selection, they first look at an initial set of relevant passages and construct a language model. Then, relevant web data is used to built a second language model. Finally, they mix the two models and include some further constraints like answer type and answer context information.

Another interesting approach is presented by Corrada-Emmanuel et al. (2003). In their paper, three methods to score relevant passages are shown. They compare the famous query likelihood, relevance modeling and a bigram answer model. In this model, the expected answer type is taken into account. Therefore, text selections are replaced by their named entity tag and an answer model is trained. Then three

different methods for backing-off the bigram are used. They show that the bigram method provides a performance superior to the other approaches.

As mentioned above, sentence retrieval is just a special case of passage retrieval where the text selection has the size of one sentence. There is also some related work in the area of sentence retrieval for QA systems. One example is Murdock and Croft (2004). They understand the meaning of retrieving sentences as the translation of a user query to a (more or less complex) answer. With this idea, they suppose to overcome the problem of the shortness of sentences to compute a multinomial distribution. Their approach is based on the IBM Model 1 and is smoothed with the corresponding document in addition to the collection. They show a performance gain to the original query-likelihood scoring.
In Losada (2005) language model based approaches for sentence retrieval are compared. They define multinomial and multiple-Bernoulli distributions on top of the query-likelihood approach. Their motivation is the shortness of a sentence. In a multiple-Bernoulli framework, also the non-query terms are taken into account. So, they show a significantly performance increase compared to a multinomial approach.

An other application for sentence retrieval is the TREC Novelty track (Harman 2002). Here, the task is to reduce the amount of redundant and non-relevant information in a given document set. Normally, this is done in a two-step approach. The first part is to find the relevant sentences according to a query[3]. In a second part, those sentences are selected which contain novel information compared to the retrieved set in the first part. Larkey et al. (2003) and Allan et al. (2003) give some examples of how to build such a system. They use three different methods for extracting relevant sentences; a vector based approach using *tf-idf*, a version using the *Kullback-Leibler divergence (KLD)* and an approach using a *Two-Stage Smoothing* model. Because they do not find any significant differences between these methods[4], they decide to use the *tf-idf* approach. From their point of view, the selection of the relevant sentences is the major challenge, so they try to further improve the performance by using known techniques like query expansion, pseudo-relevance feedback and other features. But, again in contrast to our observations, just pseudo-feedback helps to improve the performance.
A major difference to the open-domain question answering is that in the Novelty track, a set of relevant documents is given. So, there is no need to find some relevant documents out of a large corpus first. Allan et al. (2003) also show the negative effects when using a real information retrieval system instead of a given document set.
A last significant difference is the kind of processing the retrieved data. In a question answering system, further steps are the extraction and selection of possible answers out of the sentences. This task is very hard and time-consuming, so it is necessary to keep the set of returned sentences as small as possible.

A partial implementation of the system can be found in Shen et al. (2006). There, a complete statistically-inspired QA system in context of the TREC 2006

[3]This is what we call *Sentence Retrieval*.
[4]In contrast to our experiments.

question answering track is developed.

Merkel and Klakow (2007) give a more specific description of the language model based query classification part we used in our experiments. This work mainly depicts the methods of how to obtain the expected answer types.

3.3 Methods

In this section, the general idea behind language model based information retrieval is presented. Furthermore, we describe the smoothing methods we used for our experiments in Sect. 3.4.3.

3.3.1 Language Models for Sentence Retrieval

First, we want to introduce the language model based approach proposed by Ponte and Croft in 1998 (Ponte and Croft 1998) as our information retrieval framework for sentence retrieval. They rank the user query using a query model, whereas a language model for each document is determined. Then the probability of producing the query with those models is calculated. Following Zhai and Lafferty (2001), applying Bayes rule results in

$$P(D|Q) \propto P(Q|D)P(D) \tag{3.1}$$

where $P(D)$ is the prior belief of a document and $P(Q|D)$ is the probability of the query given a document.

We act on the assumption that the prior $P(D)$ is a uniform distribution, so it is equal for all documents and therefore irrelevant for ranking the query. Thus, it will be ignored in further computations. The probability of $P(Q|D)$ is calculated by using language models. This conversion means that we just have to calculate the conditional probability of the user query and the document we intend to rank. This task seems easier than calculating $P(D|Q)$.

Formula (3.1) has a data sparsity problem. Generally, there isn't enough training data to compute language models for a complete query[5]. To overcome this problem we act on the assumption that all words in the query are independent. This independence assumption results in unigram language models as proposed in Zhai and Lafferty (2001):

$$P(Q|D) = \prod_{i=1}^{N} P(q_i|D) \tag{3.2}$$

whereas N is the number of terms in a query. In our approach the documents are sentences, so we used $P(q_i|S)$ as our experimental baseline, where S is the sentence we intend to score.

In the next sections we will describe how to calculate those probabilities. Because we use a maximum likelihood estimate to calculate $P(w|S)$, it is necessary to smooth them in order to avoid zero probabilities[6].

[5] Let's suppose that a query has 7 words in average. Then 7–gram language models have to be computed.
[6] See Zhai and Lafferty (2001) for further information.

3.3.2 Jelinek–Mercer smoothing

The Jelinek-Mercer smoothing method is just a linear interpolation between the maximum likelihood probability and a background collection model. It is defined by

$$P_\lambda(w|S) = (1-\lambda)\frac{c(w,S)}{\sum_w c(w,S)} + \lambda P(w|C) \tag{3.3}$$

where $c(w,S)$ is the count of word w in sentence S and λ is the smoothing parameter. $P(w|C)$ is the collection model. In our experiments the background collection always consists of the set containing all sentences.

3.3.3 Absolute Discounting

This smoothing method has its origin in the task of speech recognition. There, it is the most efficient and thus the most commonly used technique. But it was also introduced to the task of information retrieval by Zhai and Lafferty (2001). It results in

$$P_\delta(w|S) = \frac{\max\left(c(w,S)-\delta,0\right)}{\sum_w c(w,S)} + \frac{\delta B}{\sum_w c(w,S)} P(w|C) \tag{3.4}$$

whereas $c(w,S)$ are the frequencies of w in S and $P(w|C)$ is the collection model of all sentences. δ defines the smoothing parameter to redistribute some probability mass to unseen events. The parameter B counts how often $c(w,S)$ is larger than δ.

3.3.4 Bayesian smoothing with Dirichlet priors

Bayesian smoothing using Dirichlet priors is the approach which performs best according our question answering task in document retrieval as well as according our sentence retrieval framework. It is also described by Zhai and Lafferty (2001) and is defined by

$$P_\mu(w|S) = \frac{c(w,S) + \mu P(w|C)}{\sum_w c(w,S) + \mu} \tag{3.5}$$

where $c(w,S)$ is the frequency of observations of the word w in sentence S. μ is the smoothing parameter. Again, $P(w|C)$ is the collection model containing all sentences. A special case of this method is the *add-epsilon smoothing*, i.e. when a uniform collection model is used.

3.4 Experiments

In this section, we describe the dataset and the experimental setup we used for our experiments. Furthermore, we discuss the experimental results.

3.4.1 Dataset

As dataset for our experiments we used the *TREC 2004 QA collection*. It consists of the AQUAINT[7] document collection with more than one million[8] text documents from various news agencies (the Xinhua News Service (People's Republic of China), the New York Times News Service, and the Associated Press Worldstream News Service).

The question set for TREC 2004 consists of 351 questions, which are further divided into subsets. Each subset has a unique topic and a set of "factoid", "list" and "other" question. For example, a typical "factoid" question is "When was James Dean born?" whereas a "list" question would be "What movies did James Dean appear in?". The task of the "other" question is mainly to find as many different information concerning the topic as possible.

As evaluation metrics for the results, the Mean Reciprocal Rank (MRR) and the accuracy of the system was used. In this context, accuracy means the percentage of answerable questions using a specific number of returned sentences. For testing the parameters in the query construction, we used the Mean Average Precision (MAP).

3.4.2 Experimental Setup

For efficiency reasons we chose a three-step approach for our experiments. First, the user question was analyzed. Therefore, we used the approach described in Merkel and Klakow (2007) to extract the expected answer type. It specifies a language model based query classification, using a simple Bayes classifier as paradigm. The taxonomy of the classifier takes 6 coarse and 50 fine grained classes into account.
In addition to this, some simple methods were used to further optimize the query for the following retrieval task.

In a second step, the *Lemur Toolkit for Language Modeling and Information Retrieval*[9] was used to carry out a language model based document retrieval. As suggested in Hussain et al. (2006), we performed Bayesian smoothing with Dirichlet priors. We fetched the top 50 relevant documents because Shen et al. (2006) showed that this number is sufficient to answer about 90% of questions. After the extraction, we split them up into sentences using the sentence boundary detection algorithm provided by LingPipe[10]. We also used larger passages (Hussain et al. (2006)), but the sentence-based approach is much more efficient.

The third step was the re-ranking of sentences using the language model based methods described in Sect. 3.3. For these purposes a new language modeling toolkit was developed by our chair[11]. It implements all standard language modeling techniques and is more flexible than other tools in terms of backing-off strate-

[7] `http://www.ldc.upenn.edu/Catalog/docs/LDC2002T31/`
[8] 1,033,461 documents.
[9] `http://www.lemurproject.org/`
[10] `http://www.alias-i.com/lingpipe/`
[11] *LSVLM.*

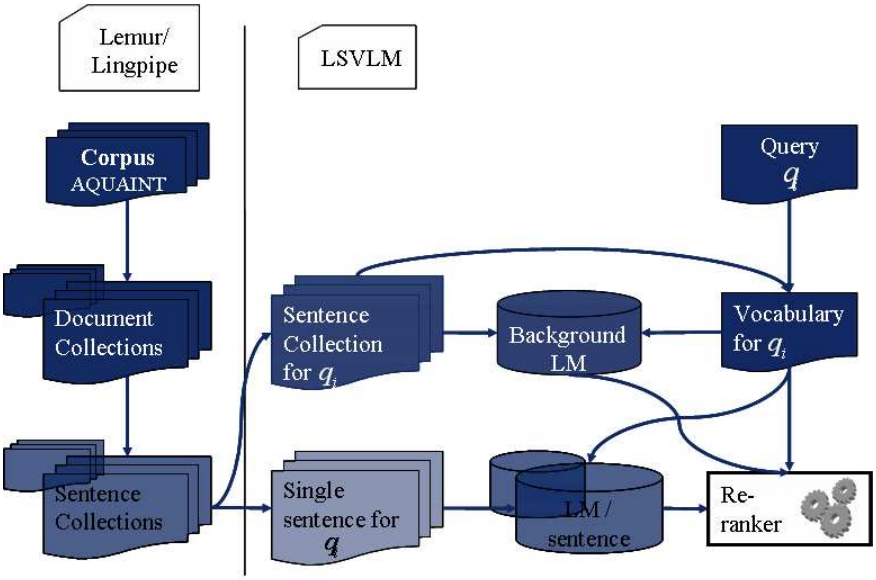


Figure 3.2: The sentence retrieval architecture for our experiments

gies, model combinations and design of the retrieval vocabulary.

Figure 3.2 shows the architecture of the sentence retrieval experiments we made. On the left-hand side, the pre-processing steps with standard software are shown. On the top one finds the Aquaint corpus we described in Sect. 3.4.1. Then the document collections, we gain by using Lemur, can be seen. On the bottom one finds the sentence collections we receive by using the LingPipe toolkit.

On the right-hand side, we show the experimental setup for the *LSVLM* framework. In the middle the background model for each experiment is presented. It consists of the complete sentence collection for a given user query q_i. Out of this collection, a background language model is build. As vocabulary for this model we use the union of the vocabulary build from the user query q_i and the corresponding sentence collection. So, the vocabulary is closed over the query.
On the bottom left, the single sentences for a query q_i can be seen. These are the sentences we want to re-score in our experiments. Then, again language models are created for each individual sentence by using the closed vocabulary.
And finally, the two language models are used to calculate a new score. Because of the flexibility of our toolkit, it is possible to easily change the used smoothing algorithms and parameters to get the optimal setting.

# included topics	Mean Average Precision
0	0.0900
1	0.2440
2	0.2995
3	0.2997
4	0.2876
6	0.2799

Table 3.1: Number of included topics and corresponding MAP for TREC 2004 dataset

In our sentence retrieval experiments, we used TF–IDF and OKAPI[12] as standard baseline approaches and linear interpolation (Jelinek-Mercer), absolute discounting and Dirichlet priors as language model based smoothing algorithms (see Sect. 3.3).
The smoothing parameters for the different methods were experimentally defined on the TREC 2003 dataset. For absolute discounting, we took the discounting parameter $\delta = 0.1$, for linear interpolation the smoothing parameter was set to $\lambda = 0.8$ and for Dirichlet prior we set $\mu = 100$.

3.4.3 Results

In this section, we discuss the results we achieved by using the query construction, the document retrieval and the optimized sentence retrieval steps.

3.4.3.1 Query Construction and Document Retrieval

As already mentioned in Sect. 3.4.2, we first analyzed the user query by extracting the expected answer type[13]. This answer type is used in a later step to optimize the language models in the sentence retrieval step.

In addition to this approach, we used further methods to optimize the query for document and sentence retrieval. In a first step, the topic of the query[14] was included for multiple times. This inclusion was done because, within our language model approach, the repeating of a specific term for multiple times results in a higher score for that term. A higher score means that the included term gets greater importance in that context.
Table 3.1 shows the impact of including the topic on the document retrieval. The performance increases until the topic was added three times. This means, if we add the topic too often, it gets too much weight and other possible relevant keywords are scored too lowly. This would result in a worse retrieval performance. Due to

[12] See `http://www.lemurproject.org`
[13] Results can be found in Merkel and Klakow (2007).
[14] See Sect. 3.4.2 for a definition of “topic”.

the fact that there is just a very small performance gain between adding the topic twice or three times, we decided to include the topic in our experiments only twice.

The last step in the query construction was the subtraction of the query word. In general, this term has no positive effects on the retrieval system and can therefore be ignored. Here, the same argumentation holds as for including the topic for multiple time. By removing the query word, this score will be zero and other, possibly more relevant terms get a higher score.

The effects of these methods on the sentence retrieval framework can be found in Sect. 3.4.3.3.

As mentioned above, the *LEMUR* toolkit was used to perform document retrieval. Therefore, the queries as well as the AQUAINT corpus were stemmed and no stop-words were removed. Then we chose a language model based approach to retrieve the documents. As smoothing method, we used Bayesian smoothing with Dirichlet priors because Hussain et al. (2006) illustrated that this approach performs best for this task[15]. They also suggests an optimal smoothing parameter for this question set which we also used for our experimental setup.
After doing the retrieval, the 50 most relevant documents were fetched and split up into sentences.

3.4.3.2 Baseline Experiments

This section describes the baseline experiments we conducted before starting our optimization approaches. Figure 3.3 shows the results of those experiments. It presents on the x-axis the number of returned sentences by the system on a logarithmic scale. On the y-axis the accuracy of the system is shown. For example, an accuracy of 0.5 means that 50% of the queries are answerable by the system.

For the standard TF–IDF and OKAPI baseline experiments we used the *LEMUR* toolkit. The figure shows that the TF–IDF performs better than OKAPI regarding this task. Both approaches were not optimized for these experiments.

In a next step, we used our *LSVLM* toolkit to conduct the baseline experiments with the three standard language model based smoothing approaches (as described in Sect. 3.4.2).

For a small number of returned sentences (1–50), the linear interpolation (Jelinek–Mercer) and the absolute discounting smoothing perform comparably bad. In this part the Bayesian smoothing with Dirichlet priors obviously performs better.
In the last segment (50–100 sentences) the Dirichlet prior approach performs somewhat worse than absolute discounting. The best smoothing method for this part is the Jelinek–Mercer interpolation. But this performance gain is not visibly significant.

But the figure also shows that all baseline language model based approaches perform better than the standard TF–IDF and OKAPI methods for this task by large margin.

[15] They show that it even provides a performance superior to standard approaches like TF–IDF and OKAPI.

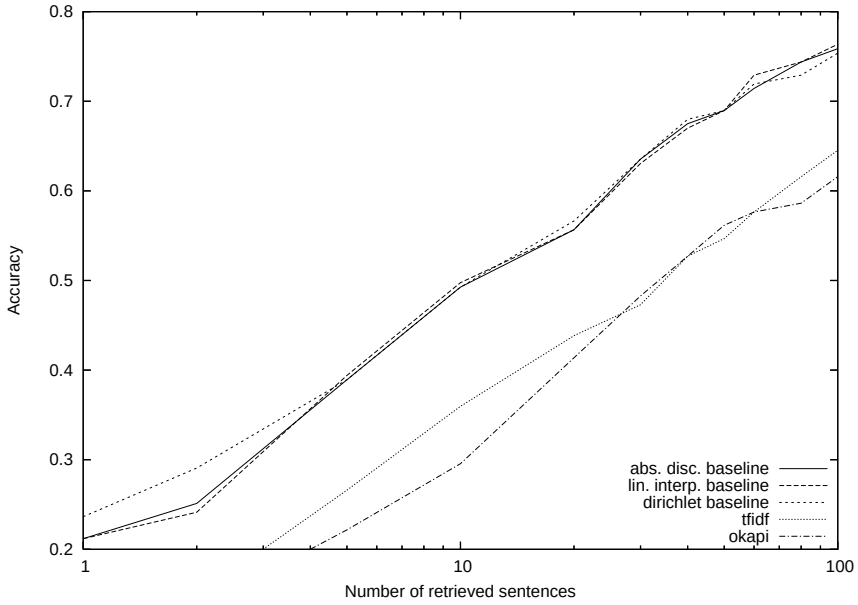


Figure 3.3: Number of retrieved sentences vs. accuracy for baseline experiments

As already mentioned in Sect. 3.1, in a question answering system we are most interested in getting a high accuracy at a small number of returned sentences. That means, the following modules need to process just smaller sets of sentences to reach the same level of accuracy. Thus, the Bayesian smoothing with Dirichlet priors was chosen as a optimization baseline for further experiments.

3.4.3.3 Improved Smoothing Methods

Figure 3.4 shows the results of the experiments we carried out with optimized language models for the question answering task. Again, on the axis of abscissae the number of returned sentences is plotted on a logarithmic scale, whereas on the ordinate the accuracy of the system is shown (as described in Sect. 3.4.1).

For better comparison between the improvements of the optimization steps, the Dirichlet prior smoothing method is shown as baseline (curve (1)). The other lines show the performance gain of each individual method we added to the baseline. Each new experiment is based on the previous optimization method.

Our first approach is already discussed in Sect. 3.4.3.1. It is the simple removal of the query word and therefore belongs to the query construction step. Figure 3.4 shows the resulting effects on the system. The new curve (2) provides a performance superior to the Dirichlet baseline.

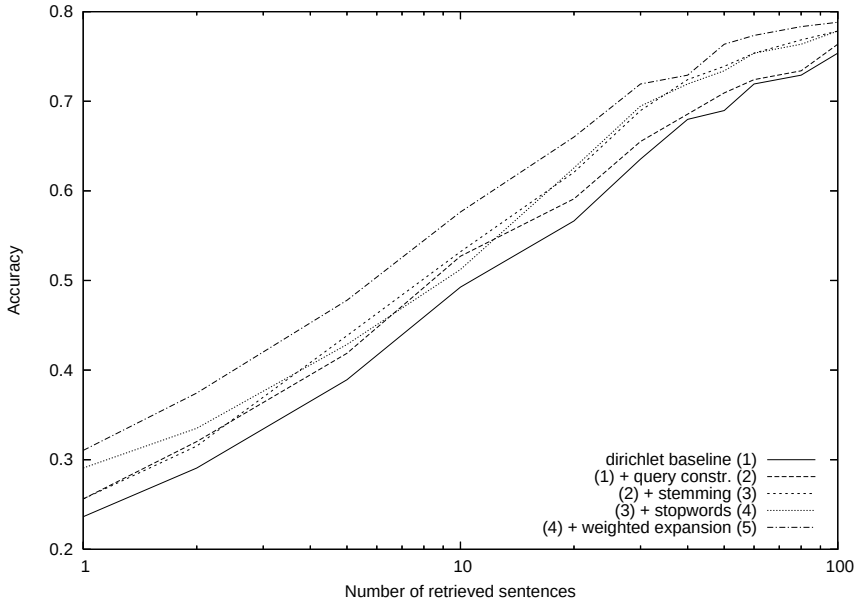


Figure 3.4: Number of retrieved sentences vs. accuracy for optimized smoothing methods

In a second step, we added the Porter stemmer[16] to our experiments. The results are shown in curve (3). As described in relevant literature, this addition also results in a further advance of system accuracy of this specific kind of task.

As a next optimization criterion we used a dynamic stopword list. It was created by selecting the four most commonly used terms of the complete sentence collection. However, those terms were not removed as usual for stop-words but they got just a smaller weight in the language model. This re-weighting is based on the same findings we already discussed in Sect. 3.4.3.1. The result in Fig. 3.4 shows in curve (4) a small performance gain, when looking at a very small number of returned sentences. Besides, the accuracy is nearly equal to the previous step.

For the last optimization experiment, the expected answer type of an user query we gained in the query construction, was used to expand the language models with this additional information (see Sect. 3.4.3.1). This was done by expanding the query and a sentence in dependency of the extracted question type. Thereby, sentences, which match the query type, are ranked higher.

This means, for example, if the expected answer type is *Date*, the term *DATE* is added to the question. Then patterns are used to identify dates expressions in a sentence. If such a date expression also occurs in a sentence, it is expanded with the *DATE* term as well. After this step, the additional terms are weighted and thus the language model based approach gives a higher rank to sentences which match

[16] `http://www.tartarus.org/~martin/PorterStemmer/`

Distribution	MRR
OKAPI	0.16
TF–IDF	0.18
Jelinek–Mercer	0.29
Absolute Discounting	0.29
Dirichlet Baseline	0.31
Dirichlet Combined	0.39

Table 3.2: Mean Reciprocal Rank of baseline and optimized experiments

the corresponding question.

The resulting effects of this last optimization step is also shown in Fig. 3.4. Here, curve (5) demonstrates the improvement of performance by adding the weighted expansion. The distribution outperforms all other combined methods by a large margin.

Table 3.2 shows the MRR of the baseline experiments and combination of all optimization steps. Standard OKAPI and TF–IDF achieved the worst MRR. The Jelinek–Mercer interpolation and absolute discounting baseline perform better with a MRR of 0.29. We found out that the Dirichlet prior baseline again performs a little bit better with a MRR of 0.31. This was the reason for the fact why we developed the improved language models on top of this distribution. The table also shows that the combination of all optimization steps (Dirichlet Combined) performs best with a MRR of 0.39. This means that there is an improvement of more than 25% compared to the Dirichlet baseline and, that there is an improvement of more than 34% compared to the other LM based experiments.

3.5 Conclusion

In this paper, we showed a language model based framework to perform improved sentence retrieval in a question answering context. The major goal was to improve the accuracy of the system in order to return just a smaller number of relevant sentences. This reduces the search space of the following components in a QA system. Because these components are typically deep-analysis approaches which strongly depend on the size of processing documents, such a step is necessary.

For this purpose, we first analyzed the user query by extracting the expected answer type and doing some other simple text manipulations.
After a language model based document retrieval step, we split up the documents into smaller text passages in the size of sentences.

Then, the *LSVLM* toolkit, a language model based framework we developed at our department, was introduced. With this toolkit, we were able to conduct sentence retrieval experiments in a more flexible way than with other state-of-the-art information retrieval frameworks. We conduct baseline experiments with standard TF–IDF and OKAPI as well as with language model based smoothing methods

like Jelinek–Mercer interpolation, Bayesian smoothing with Dirichlet priors and absolute discounting.

We proved that Dirichlet priors baseline performs best for our task, so we developed our optimization steps on top of this approach.
In several experiments we illustrated that using query word removal, dynamic stopword list weighting and stemming results in a performance gain. In the last experiment, we modeled the expected answer type of a user query into the used language models. This approach performs better than the LM baselines by at least 25%.

We also proved that we need to return fewer sentences in order to achieve equal or even better performance in terms of system accuracy. So we attained our goal to reduce the search space for the following components in a QA framework.

References

Allan, J., Wade C., and Bolivar, A. (2003). Retrieval and Novelty Detection at the Sentence Level, *in Proceedings of the Special Interest Group on Information Retrieval (SIGIR) 2003*, Toronto, 2003.

Clarke, C.L.A., Cormack, G.V., Kisman, D.I.E., and Lyman, T.R. (2000). Question Answering by Passage Selection (MultiText experiments for TREC 9), *in The 9th Text REtrieval Conference (TREC-9)*, Gaithersburg: NIST, 2000.

Corrada-Emmanuel, A., Croft, W.B., and Murdock, V. (2003). Answer Passage Retrieval for Question Answering, *in CIIR Technical Report*, Amherst, 2003.

Harman, D. (2002). Overview of the TREC 2002 Novelty Track, *in The Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg: NIST, 2002, pp. 17–28.

Hussain, M., Merkel, A., and Klakow, D.(2006). Dedicated Backing-Off Distributions for Language Model Based Passage Retrieval, *in Hildesheimer Informatik-Berichte, LWA 2006*, Hildesheim, 2006.

Larkey, L.S., Allan, J., Connell, M.E., Bolivar, A., and Wade, C. (2003). UMass at TREC 2002: Cross Language and Novelty Tracks, *in The Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg: NIST, 2003.

Losada, D.E. (2005). Language Modeling for Sentence Retrieval: A Comparison between Multiple-Bernoulli Models and Multinomial Models, *in Information Retrieval and Theory Workshop*, Glasgow, 2005.

Merkel, A. and Klakow D. (2007). Language Model Based Query Classification, *to appear in Proceedings of 29th European Conference on Information Retrieval (ECIR)*, Rome, 2007.

Murdock, V. and Croft, W.B. (2004). Simple Translation Model for Sentence Retrieval in Factoid Question Answering, *in Proceedings of the Special Interest Group on Information Retrieval (SIGIR) 2004*, Sheffield, 2004.

Ponte, J.M. and Croft, B. (1998). A Language Modeling Approach to Information Retrieval, *in Proceedings of the Special Interest Group on Information*

Retrieval (SIGIR) 1998, Melbourne, 1998.

Shen, D., Leidner, J.L., Merkel, A., Klakow, D. (2006). The *Alyssa* System at TREC 2006: A Statistically-Inspired Question Answering System, *in The Fifteenth Text REtrieval Conference (TREC 2006)*, Gaithersburg: NIST, 2006.

Tellex, S., Katz, B., Lin, J., Fernandes, A., and Marton G. (2003). Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering, *in Proceedings of the Special Interest Group on Information Retrieval (SIGIR) 2003*, Toronto, 2003.

Zhai, C. and Lafferty J. (2001). A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval, *in Proceedings of the Special Interest Group on Information Retrieval (SIGIR) 2001*, New Orleans, 2001.

Zhang, D. and Lee, W.S. (2004). A Language Modeling Approach to Passage Question Answering, *in NIST Special Publication 500–255: The Twelfth Text REtrieval Conference (TREC 2003)*, Gaithersburg: NIST, 2004.

4

Conditional Entropy Measures Intelligibility among Related Languages

Jens Moberg[†], *Charlotte Gooskens*[†], *John Nerbonne*[†], *and Nathan Vaillette*[‡]
[†]University of Groningen
[‡]Dickinson College, Pennsylvania

Abstract

The Scandinavian languages are so alike that their speakers often communicate, each using their own language, which Haugen (1966) dubbed SEMICOMMUNICATION. The success of semi-communication depends on the languages involved, and, moreover, can be asymmetric: for example, Swedish is more easily understandable for a Dane, than Danish for a Swede. It has been argued that non-linguistic factors could explain intelligibility, including its asymmetry. Gooskens (2006), however, found a high correlation between linguistic distance and intelligibility. This suggests that we need to seek linguistic factors that influence intelligibility, and that potentially asymmetric factors would be particularly interesting. Gooskens' distance techniques cannot capture asymmetry. The present paper attempts to develop a model of the success of semi-communication based on conditional entropy, in particular using the conditional entropy of the phoneme mapping in corresponding (cognate) words. Semantically corresponding words were taken from frequency lists and aligned, and the conditional entropy of the phoneme mapping in aligned word pairs was calculated. This gives us information about the difficulty of predicting a phoneme in a native language given

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

a corresponding phoneme in the foreign language. We also examine the conditional entropy of selected word classes, such as native/loan and function/content words.

4.1 Introduction

The three mainland Scandinavian languages (Danish, Norwegian and Swedish) constitute an interesting linguistic community with respect to mutual intelligibility. They are so closely related that they are sometimes considered dialects of a common, non-existent, language (Maurud 1976, Braunmüller 2002). This linguistic situation enables citizens in Scandinavia to use their native tongues when communicating with their neighbors. Haugen (1966) coined the term SEMICOMMUNICATION for this phenomenon, for which Braunmüller (2002) suggests rather RECEPTIVE MULTILINGUALISM.

4.1.1 Background

It has been noted that semicommunication may be difficult, and several studies, the most prominent being Maurud (1976), Bø (1978), and Delsing and Lundin Åkesson (2005), were carried out in order to investigate how well speakers of the three languages understand the neighboring languages. We calculated the mean percentage of correct answers in the intelligibility tests of these three investigations, and display these per language pair in Figure 4.1. The largest problems are found in the mutual intelligibility between Swedes and Danes. Swedes especially have difficulties understanding Danish (a mean of 27% correct answers as opposed to 37% correct when Danes attempt to understand Swedish). Norwegians understand the neighboring languages best, while Danes and Swedes both have more difficulties understanding Norwegian.

Intelligibility is asymmetric in all of the language pairs in Fig. 4.1, and intelligibility scores are often explained by appeals to attitude and amount of contact. A positive attitude should encourage subjects to try to understand the language in question, whereas a negative attitude will discourage subjects from making an effort. Contact with the language in its written or spoken form is also likely to improve the performance on the test. The good performance by the Norwegians may be explained by the fact that the language variety of the listeners (eastern Norwegians) is linguistically close to both Danish and Swedish, Furthermore, it has been proposed that Norwegians are particularly good at understanding closely related language varieites because the Norwegian dialects are used so extensively. In contrast to many European countries dialects are used by people of all ages and social backgrounds in Norway, not only in the private domain but also in official contexts (Omdal 1995). For this reason Norwegians are used to decoding different language varieties. The influence of this factor on semicommunication has, however, never been tested experimentally. The three Scandinavian studies mentioned above included questions about attitude towards and contact with the test language. The authors assume a relationship between the non-linguistic factors (attitude and experience) and the intelligibility scores, but correlations are low

and the direct relationship is difficult to prove. A third factor, linguistic structure, has been largely neglected so far, mostly due to the absence of a suitable method to measure differences in linguistic structure. In recent years, new methods have been developed for measuring linguistic differences in the area of dialectometry. This makes it possible to measure communicatively relevant linguistic differences among the spoken Scandinavian languages. Linguistic differences can be measured at various linguistic levels, but we shall be concerned exclusively with the phonetic level in this paper.

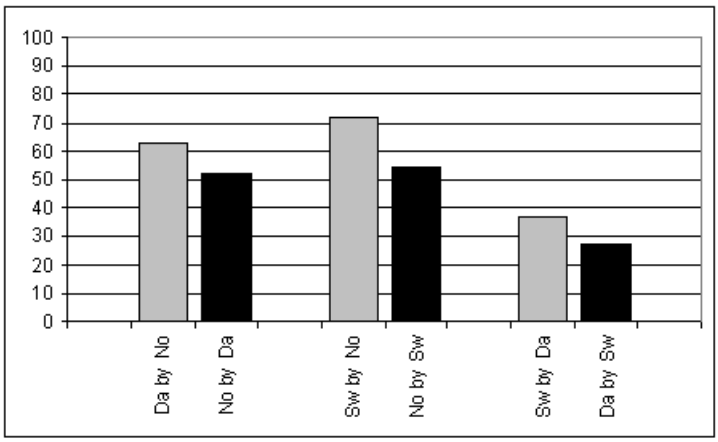


Figure 4.1: Mean percentage correct answers of three spoken intelligibility tests (Maurud 1976, Bø 1978, Delsing & Åkesson 2005). 'Da by No' stands for 'Percentage correct in Danish test by Norwegians', etc.

Heeringa (2004, Chap. 7–8) describes a method for measuring the phonetic distance between dialects and closely related languages by means of the Levenshtein algorithm. This algorithm calculates the minimum cost of transforming one sequence of phonemes to another. Gooskens (2006) used these distance measurements, and found a high correlation between intelligibility and phonetic similarity measured by means of Levenshtein distances ($r = 0.82$, $p \leq 0.01$). However, since the Levenshtein algorithm calculates distances, which are axiomatically symmetric, it cannot provide an account of asymmetric relations in linguistic intelligibility.

4.1.2 Present Paper

The present paper explores the linguistic differences among the Scandinavian languages by means of another measure, conditional entropy, which we apply at the phonemic level. Conditional entropy measures the complexity of a mapping, and is sensitive to the frequency and regularity of sound correspondences between two

languages. Since these two factors could be important to the ease with which a word in one language is understood by speakers of a related language, we hypothesize that conditional entropy corresponds with intelligibility scores. We are motivated to explore conditional entropy because it can model asymmetric remoteness. The conditional entropy between language A and language B is not necessarily the same as between language B and language A. If the asymmetric intelligibility scores reported above (see Figure 4.1) reflect the difficulty of mapping one sound system to another, we may expect conditional entropies to operationalize this difficulty, so that high entropies correspond with low intelligibility between a given pair of Scandinavian languages, and low entropies with high intelligibility. The primary purpose of this paper is to test this hypothesis.

We based our measurements on a database with frequent words in the three languages. This database was divided into different categories which made it possible to test three hypotheses. First, we expected native words to produce a higher conditional entropy between pairs of languages than loan words, since they have evolved in the respective languages for a long time. Loan words entering a language are expected to differ less because they have been borrowed in a similar form and have not had the time to diverge as much.

Second, we expected lower entropies for Latin/Greek/French loan words than for German loan words because the time of borrowing differs. Most German loan words came into Scandinavian in the twelfth and thirteenth century, during the Hanseatic period. French loan words became popular in the sixteenth century (Edlund and Hene 1992). Words imported into the Scandinavian languages were often adapted in some way during the process. Assume that in a borrowed word, sound A becomes sound B in Swedish, sound C in Danish and sound D in Norway. The way that the Scandinavian languages transform this sound to fit their own language is to a certain degree a regular process, meaning that the pairwise relations (between B and C, etc.) are rule governed. However, since the German loan words have been part of the Scandinavian languages for a longer time, they have had more time to change, which means that the regularities may have attenuated. The fact that French, Latin and Greek are less closely related to the Scandinavian languages than German might also mean that the words have been less well integrated into the Scandinavian languages than German words. For this reason the Latin/Greek/French words may to a greater extent have kept their original pronunciation. This might cause lower conditional entropies for Latin/Greek/French loan words than for German loan words.

Third, we make a distinction between function words and content words. Many function words are very frequent, and since they are less essential to the semantic content of sentences, they often occur in unstressed positions. For this reason their form may have been more strongly reduced than that of content words. In addition, very frequent words are also said to be phonologically conservative, i.e. they resist regular changes. Both observations lead us to expect conditional entropies to be higher for function words than for general vocabulary, since in both case they may represent exceptions to rules.

To summarize, our specific research questions are as follows.

1. Do high conditional entropies correspond to low intelligibility scores as found in the literature and *vice versa* (see Figure 4.1)?
2. Can asymmetric mutual intelligibility be modeled by conditional entropies?
3. Is there a difference in conditional entropies between native words and loan words?
4. Is there a difference in conditional entropies between Latin/Greek/French loan words and German loan words?
5. Is there a difference in conditional entropies between content words and function words?

4.2 Conditional entropy

Conditional entropy (CE) measures the entropy, or uncertainty in a random variable when another is known. In the case we have in mind, an interlocutor hears a phoneme in a non-native language and attempts to map it to a phoneme in his own. The conditioning variable is the phoneme heard in the non-native language, and the conditioned variable is the phoneme to be identified.

Conditional entropy is calculated with the following formula:

$$H(X|Y) = -\sum_{x \in X, y \in Y} p(x,y) \log_2 p(x|y) \tag{4.1}$$

As the formula clarifies, CE is always calculated on the basis of the conditional probability of one variable given another.

$H(X|Y)$ is the uncertainty in X given knowledge of Y, i.e. how much entropy remains in X if the value of the variable Y is known. We use CE to measure the uncertainty, and therefore difficulty of predicting a unit in the native language given a corresponding unit in the non-native language.

We note that CE is asymmetric, i.e. it does not hold in general that $H(X|Y) = H(Y|X)$. This means that it will not run into the same conceptual difficulties as the distances used by Gooskens (2006).

4.2.1 Plausibility

As a simplest illustration of how conditional entropy can be used for linguistic units, consider the following. Written Danish words have only one vowel in their grammatical endings, the letter *e*, while Swedish uses *e*, *a* and *o*. This means that a Swedish speaker that encounters the Danish letter *e* has three options when trying to find the equivalent Swedish phoneme. Idealizing now to the situation where this were the only use of the sounds in question, we can see that a Danish speaker, upon encountering Swedish *e*, *a* or *o*, can know that the proper correspondence is *e*. The entropy is therefore higher for Swedish given Danish in this example, and the relationship is asymmetric.

Table 4.1: Corpus of Two Phonetically Transcribed Word Pairs

Danish	Swedish
j ɑ i	j ɑː g
l ɑ ŋ ʔ	l ɔ ŋ #

4.2.2 Example: CE for 2 Danish-Swedish Word Pairs

If the imaginary example of the perfect three-way split in the mapping serves to motivate the idea of using the complexity of the mapping as a model for intelligibility, it suffers from being too simple and from not taking frequency into account. It is too simple in that it is seldom, if ever, the case that one sound is mapping into three (or $n \neq 0$) others, each of which participates in no other mapping. And the measure of complexity intuitively ought to involve frequency—we can also understand more easily if we have a reasonable "guess" about the correspondence, and that guess may be well informed by frequency.

We shore up this intuition using a slightly larger example, with sound segments from two aligned word pairs to calculate the aggregate conditional entropy.

Table 4.1 shows a made-up corpus containing two word pairs with a total of 13 occurrences of sound segments. The sound segments are aligned, mimicking the way a non-native interlocutor might attempt to map a foreign word to one in his own language: /j/ with /j/, /ɑ/ with /ɑː/, /i/ with /g/ and so forth. In the last word pair, Danish glottal stop is aligned with a filler symbol. The frequencies are used to estimate the probabilities needed to calculate conditional entropy (4.1), including $P(d)$, the chance of segment d occurring in Danish; $P(s)$, the chance of s in Swedish; $P(d|s)$, the chance of d in alignment, given s; $P(s|d)$, the converse; and $P(d, s)$, the chance of d and s occurring jointly (in alignment). $P(d, s)$ is used to weight the importance of the conditional probabilities $P(d|s)$ and $P(s|d)$ in the CE formula (4.1).

We illustrate how the conditional entropies would be calculated on the basis of a corpus using the data of Table 4.1 by keeping track of the alignments, including the partial alignments. We thus first align all of the data, obtaining the alignments shown in Table 4.2, which we now discuss.

In the second cell alignment in Table 4.2, Swedish /ɑː/ is matched with Danish /ɑ/. Swedish /ɑː/ occurs only once, so that $P(\text{ɑ}_D|\text{ɑː}_S)$ is therefore 1. Since $-\log 1 = 0$, this contributes nothing to entropy. In the other direction, $P(\text{ɑː}_S|\text{ɑ}_D) = 0.5$: Danish /ɑ/ corresponds to Swedish /ɑː/ in the second word pair and to Swedish /ɔ/ in the second word pair (cell 5). This type of correspondence is the cause of asymmetry in the phoneme mapping complexity: the uncertainty is higher for Swedish speakers because they have more sound segments to choose from than Danish speakers.

All the Swedish segments map uniquely to Danish counterparts so that $\forall s \in$

Table 4.2: Seven Illustrative Segment Alignments and Corresponding Frequencies. From aligned data (as shown), we extract the relative frequencies of the correspondences. The 1:1 frequencies indicate perfect correspondences, therefore conditional probabilities of 1, which correspond to zero contributions to entropy ($-\log_2 1 = 0$). Note that all of the relative frequencies marked with 'S' are perfect (1:1), so that $H(\text{Danish}|\text{Swedish}) = 0$, reflecting the perfect predictability of the Swedish → Danish mapping. 'D(1:2)' in the top row center (cell 2) indicates e.g. that, Danish /ɑ/ is realized in the way indicated in the cell (ɑː) once out of a total of two occurrences (the other is in the bottom row, second position, cell 5). The boldfaced asymmetric alignment frequencies (in cells 2 and 5) contribute to the entropy difference, $H(D|S) = 0.0 < H(S|D) = 0.28$ (in this example set).

Language	1	2	3	
D→	j	ɑ	i	
S→	j	ɑː	g	
	S(1:1), D(1:1)	S(1:1), **D(1:2)**	S(1:1), D(1:1)	
	4	5	6	
D→	l	ɑ	ŋ	ʔ
S→	l	ɔ	ŋ	#
	S(1:1), D(1:1)	S(1:1), **D(1:2)**	S(1:1), D(1:1)	S(1:1), D(1:1)

$Sp(d|s) = 1, -\log_2 p(d|s) = 0$, and the total entropy is zero, corresponding to the perfectly certain mapping. Similarly, five Danish segments map uniquely to Swedish segments, likewise contributing zero to entropy. But one Danish segment /ɑ/ is mapping 50% of the time to Swedish /ɑː/ and 50% of the time to Swedish /ɔ/. We therefore estimate that $p(\text{ɑː}|\text{ɑ}) = p(\text{ɔ}|\text{ɑ}) = 0.5$, and therefore that $-\log_2 p(\text{ɑː}|\text{ɑ}) = -\log_2 p(\text{ɔ}|\text{ɑ}) = 1$, and we use $p(\text{ɑː,ɑ}) = p(\text{ɔ,ɑ}) = 1/7 \approx 0.14$ to weight these contributions to entropy, obtaining $H(\text{Swedish}|\text{Danish}) = 2 \times 0.14 = 0.28$.

Based on the mini-corpus in Table 4.2 $H(S|D) > H(D|S)$ because of the larger number of less certain mappings (in this case the second and fifth elements of the alignments, just discussed). We hypothesize that this is true in general, and that it contributes to the lesser intelligibility of Danish for Swedes.

We turned out to need about 800 words to obtain stable estimations of phoneme mapping entropies, but smaller samples consistently gave good estimations of the relative difference between $H(\text{Lg}_1|\text{Lg}_2)$ and $H(\text{Lg}_2|\text{Lg}_1)$. See Fig. 4.2.

4.3 Material

In order to conduct the entropy measurements, a database containing word lists from the three Scandinavian languages was constructed. The database also contained the same lists in Dutch, Low German, High German and Frisian since we plan to extent our research to these languages as well. The database contains the

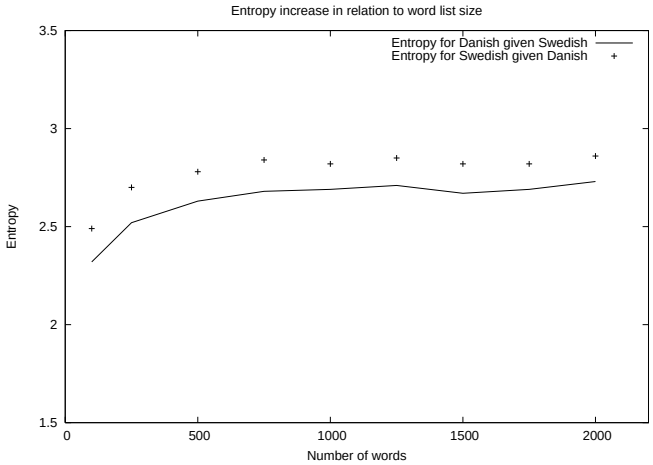


Figure 4.2: Entropy in relation to word list size

most frequent words from two corpora, Corpus Gesproken Nederlands, CGN, and Europarl[1].

CGN is a Dutch corpus of contemporary Dutch as spoken by adults in Flanders and the Netherlands that was collected between 1998 and 2004. This part of the CGN contains a total of 2,626,172 tokens. From this corpus we extracted the 1,500 most frequent words from the category that contained informal speech (the Face-To-Face dialogues).

Europarl is a speech corpus that consists of extracts from meetings held within the European Parliament. They are characterized by monologues by different speakers, including the chairman of the meeting. Europarl is translated into eleven European languages. Our motivation for choosing to extract the Dutch and Swedish version was twofold: firstly, these two languages represent the two Germanic branches of the language tree, West Germanic and North Germanic, that the database was intended to reflect. Secondly, the two languages were part-of-speech tagged, in contrast to for example Danish. We selected the 1,500 most frequent Dutch words from the Europarl, from a total of 889,836 tokens, and the 1500 most frequent Swedish words, from a total of 1,032,144 words. Next the Dutch and the Swedish lists were matched to find the 1,500 most frequent words that are common in the two lists.

The CGN list and the Europarl list were joined, and doublets were removed. The database was later supplemented with function words collected from grammar books. The goal was to make the collection of function words as comprehensive

[1] `http://lands.let.kun.nl/cgn/home.htm` and `http://www.statmt.org/europarl`, both accessed Dec 14, 2006.

as possible. Proper nouns and interjections were removed.

We based the word lists on formal as well as informal speech in order to check for differences regarding the number of loan words for these two categories. Recall that we expect words of common Northern Germanic origin to have been in the individual languages longest, followed by the words borrowed from German, followed by late borrowings from Latin, Greek and French. Recall, too, that the oldest words have the most time to undergo language-specific changes, meaning that they will be less parallel, and contribute therefore more to conditional entropies. In order to be able to investigate this hypothesis we had to ensure that a sufficient number of native words and loan words from different languages were present in our word lists. Europarl contains many words from the domains of politics and economy. These words are often borrowed from French, Latin or Greek (Gooskens et al. submitted, 2007). CGN's informal speech is collected from everyday speech situations where we expect more native words.

The material was translated so that we got word lists of the same words in the seven Germanic languages. All words were transcribed according to standardized speech as in pronunciation dictionaries, but we made no effort to verify that the standard pronunciation was in fact used in the utterances. We also looked up all the words in etymological dictionaries in order to establish from which language the loan words have been borrowed. The final version of the database contains the following information per word and language:

- The corpus from which the word was collected
- Word class
- Function word/content word
- The origin of the word (loan word or native word)
- If the word is a loan word, the language from which the word has been borrowed
- The lexical representation of the word
- The phonetic representation of the word
- Cognate/non-cognate[2] (if a word from one language is a cognate with a word from another language, these words are effectively coindexed at this feature in the database)

On the basis of this information we divided the database into a number of categories, facilitating the calculation of conditional entropies for given subvocabularies of language pairs. These categories, and their sizes in each of the three languages, are shown in Figure 4.3. The grammatical division into function

[2]The term COGNATE is usually reserved for (native) words from different languages that have descended from a common ancestor. We use an extended sense of this term to include as well words in different languages that have a common source via borrowing.

words/content words was only done for the native words since almost all function words are found in that category.

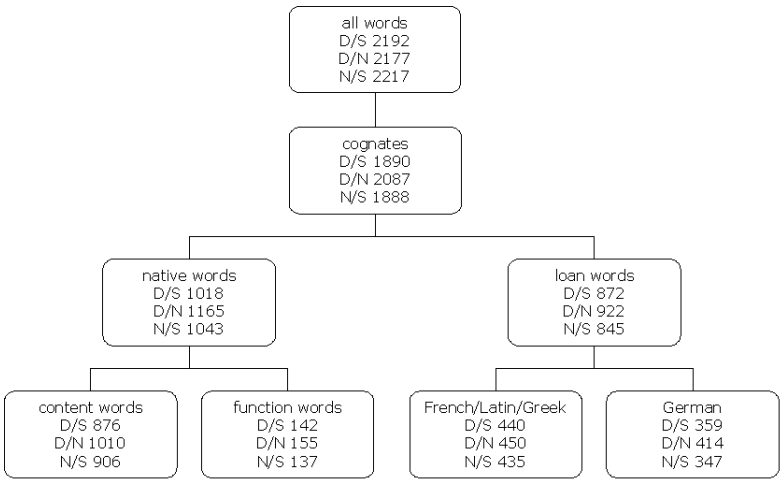


Figure 4.3: Number of Word Pairs for each of the 8 Categories.

4.4 Results

In this section we present the conditional entropies for each of the language pairs measured in both directions and compare them to the mean results of intelligibility tests presented in Figure 4.1. We look at entropies based on the entire word lists as well as subgroups containing different sub-vocabularies (see Section 4.3). To repeat, a low conditional entropy value $H(\text{Native}|\text{Foreign})$ means that mapping from the foreign language to a given native language is relatively simple: correspondences are regular and frequent. Therefore a low conditional entropy is hypothesized to correspond to high intelligibility. On the other hand, a high entropy value means a high level of uncertainty for the listener and a low level of intelligibility.

4.4.1 Danish/Swedish

Since the results of intelligibility tests show that Danes understand Swedes better than vice versa (see Figure 4.1), we expect $H(D|S) < H(S|D)$, i.e. it is less complex to map from Swedish to Danish than it is to map from Danish to Swedish.

Figure 4.4 shows the entropy per category and the divergence from symmetry. The X axis shows the entropy for Danish given Swedish, i.e. the difficulty for a

Swede to predict the Swedish equivalent of a given Danish sound segment. The Y axis shows the entropy for Swedish given Danish. The diagonal represents the completely symmetric situation, where the entropy is the same in both directions. Symbols above the line are categories which have a higher entropy for Swedish given Danish (more difficult for a Swede) while symbols under the line have a higher entropy for Danish given Swedish (more difficult for a Dane).

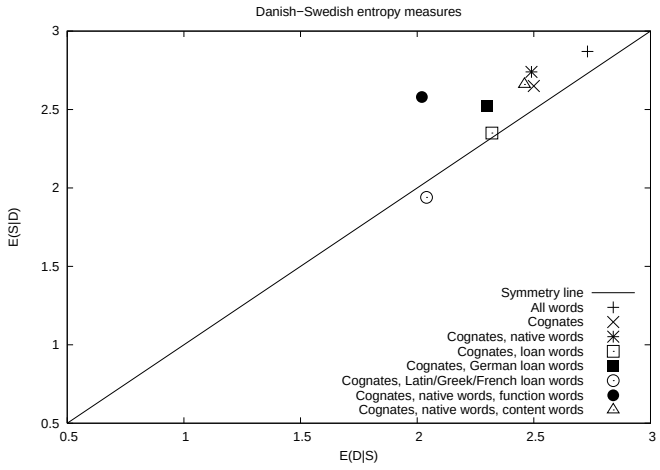


Figure 4.4: Conditional Entropy between Swedish and Danish, noting Asymmetry.

The first conclusion one can draw from these results is that there really is a difference in entropy between Swedish and Danish depending on the direction. For all categories of words, except for Latin/Greek/French cognates, the entropies are higher for Swedes listening to Danes than the other way round. This is what we would expect from the intelligibility tests (Figure 4.1).

As expected, the category consisting of all words has the highest entropy. This can be explained by the fact that this category is the biggest (see Fig. 4.2), and that it contains cognates as well as non-cognates that have no regular sound correspondences. But the group containing only cognates also has high entropy. This could be expected because it contains words of different origin, native as well as loan words, with different sound correspondences. However, when comparing native cognates to cognate loan words, we see that the native words have a higher entropy. These words have had more time to diverge than the loan words and this results in less regular sound correspondences. We see that category containing Latin/Greek/French loan words indeed have lower entropies than the German loan words. This confirms our expectation (see Section 4.1) that the correspondences for these words are more regular due to their later time of borrowing and low level of integration into the languages. Assuming that the words had more or less the same appearance in Danish and Swedish at the time of the borrowing, they have

had little time to diverge along with the respective pronunciation schemas in these countries, which in turn means more regular correspondence and lower entropy.

Contrary to our expectations, the function words have lower entropies than content words. It is possible that this can be explained by the fact that this group consists of so few words in comparison with the other groups (see Fig. 4.2).

4.4.2 Norwegian/Swedish

Figure 4.5 shows the entropy and asymmetry for the Norwegian/Swedish language pair. From the results in Figure 4.1 we expect lower overall entropies than for Swedish/Danish and we also expect the entropies to be higher for Swedish listeners than for Norwegian listeners. Both expectations are fulfilled. The Swedish/Danish entropies for the entire sample ranged between 1.94 (non-Germanic borrowings) to 2.87 (overall) bits while the corresponding Swedish/Norwegian entropies are lower, between 0.87 (non-Germanic borrowings) and 2.28 (overall). In each category of word tested, we found higher entropies and therefore more complex mappings in the Swedish/Danish case than in Swedish/Norwegian. Turning to the second expectation, it also turns out that the Swedish to Norwegian mapping is simpler than the reverse, not only overall, but also in all subcategories of words we examined (with the single exception of the category of non-Germanic loan words, where the Norwegian to Swedish mapping was slightly simpler (0.05 bits)). Among Scandinavian cognates, German borrowings, function words and content words we find lower entropies for the Swedish to Norwegian mapping.

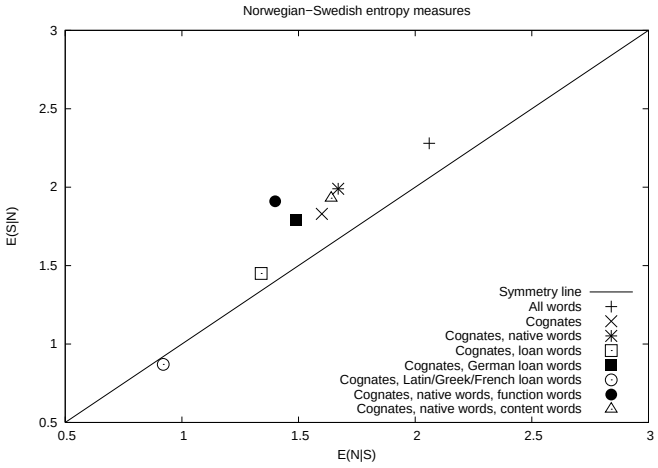


Figure 4.5: Conditional Entropy between Norwegian and Swedish

4.4.3 Danish/Norwegian

Figure 4.6 shows the entropy and asymmetry for the Danish/Norwegian language pair. The overall entropies are higher than for Swedish/Danish but lower than for Swedish/Norwegian. This corresponds with the results of the intelligibility experiments in Figure 4.1. The range between the different categories is not very large (values between 1.90 and 2.46). This can probably be explained by the fact that Danish and Norwegian have a long common history and the east Norwegian variety which represents standard Norwegian in the phonetic transcriptions has had particularly strong influence from Danish until a hundred years ago. This means that the languages were still one language when the loan words were introduced into the languages. This goes for Latin/Greek/French as well as for German loan words and therefore the entropies of these two categories are almost the same. Also the entropies of the cognate native words and the loan words are rather close. Almost all categories are close to the symmetry line. This seems to suggest that the asymmetric mutual intelligibility found in Figure 4.1 can only to a small extent be explained by differences in entropy. We will return to this in Section 4.5.

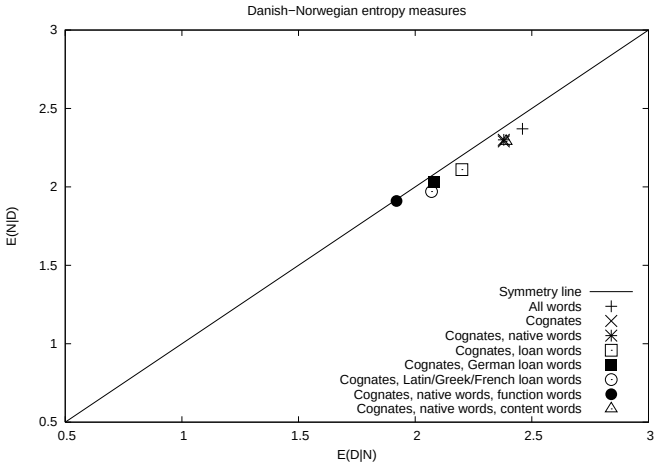


Figure 4.6: Conditional Entropy between Danish and Norwegian

4.5 Conclusions and discussion

The purpose of the present investigation was to explore conditional entropy as a linguistic measure for modeling the mutual intelligibility between closely related languages. Such a measure should also be able to model asymmetric intelligibility between for example Swedish and Danish. In Figure 4.7 we present a scatterplot which shows the relation between scores on intelligibility tests as found in the lit-

erature about semicommunication in Scandinavia (see Figure 4.1) and conditional entropies based on all words, cognates and non-cognates (circles) and on cognates only (triangular shapes). This figure clearly suggests that conditional entropies correspond well with the results of intelligibility tests. The relationship is clearest when all words are included. When the listeners were tested in the intelligibility tests they were also confronted with all words. The measurements based on cognates express pure phonetic measures of difference. Here the relationship with intelligibility scores is less clear, especially due to the fact that the two Norwegian-Danish intelligibility measures are higher than could be expected from the phonetic distances (Fig. 4.7). This might be explained by the small number of non-cognates between Danish and Norwegian (recall Figure 4.3).

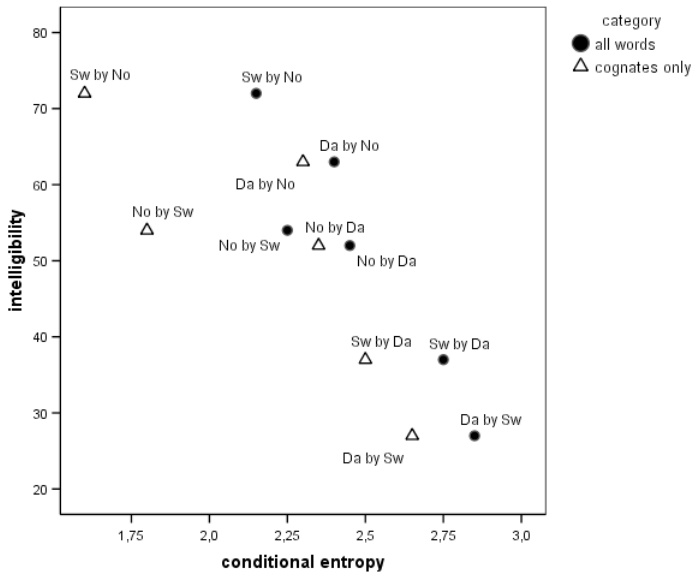


Figure 4.7: Entropy in Relation to Intelligibility

An important motivation for using conditional entropy as a measure of remoteness was that this measure is able to model asymmetric intelligibility. Asymmetric intelligibility is found between all Scandinavian language pairs (see Figure 4.1). This asymmetry was clearly reflected in the conditional entropies of Swedish-Danish (see Figure 4.4) and Swedish-Norwegian (Figure 4.5), but only to a small degree for Danish-Norwegian (Fig. 4.6). As mentioned in Section 4.1, the fact that Norwegians are better at understanding the neighboring languages than Danes and Swedes is mostly explained by the special Norwegian language situation that trains the Norwegians to understand different language varieties. The asymmetric intelligibility is larger for Swedish-Norwegian than for Danish-Norwegian. So maybe

for Swedish-Norwegian, the asymmetry is caused by a combination of language experience and linguistic factors while for Danish-Norwegian linguistic factors play only a minor role.

For all language pairs we found lower entropies for loan words than for native words. We explained this by the fact that loan words have had less time to diverge than native words which has resulted in more regular sound correspondences in the loan words. This explanation is supported by the fact that German loan words have higher entropies than the more recently borrowed Latin/Greek/French loan words. The fact that loan words have lower entropies than native words seem to suggest that a large number of loan words may benefit the intelligibility between the Scandinavian languages. This is an interesting prospect. The worry of linguistic deterioration as a consequence of too many loan words might be toned down if it turns out that loan words favor mutual intelligibility. The idea of having a common Scandinavian policy for acceptance of loan words could also find support in this result.

In future research we will refine the entropy model in several ways. More sophisticated measures will be developed that are able to express the fact that for example consonants are more important for decoding cognates than vowels and that not all phonotactic positions are of equal importance for understanding. The onset is clearly the most important position at least within the Germanic language family. We will also experiment with measurements based on bigrams or trigrams. Mutual intelligibility in Scandinavia is well documented so that the Scandinavian languages formed a good point of departure for our measurement. Our corpus contains more Germanic languages and we will apply our measurement to these languages as well. On this note, we have recently begun a collaboration with colleagues in Nijmegen and Leuven on comprehensibility among various Dutch varieties in the Netherlands and Flanders. Furthermore, we will collect material from other languages pairs which are known to have asymmetric mutual intelligibility, for example Spanish-Portuguese. At present we conclude only that there seems to be a relationship between entropy and the intelligibility experiments reported in the literature. To be more certain we need to conduct intelligibility experiments testing the hypotheses under controlled circumstances.

Acknowledgments

We are grateful to the Volkswagen Foundation, whom we thank for their support of "Measuring Linguistic Unity and Diversity", P.I. Erhard Hinrichs, Tübingen, and also the Dutch Organization for Scientific Research, who fund "Linguistic Determinants of Mutual Intelligibility in Scandinavia", P.I. Charlotte Gooskens. We are also grateful to colleagues T. Zastrow of Tübingen; P. Osenova, K. Simov, V. Zhobov of Sofia; J. Prokić of Groningen; and to two anonymous CLIN referees for discussion and suggestions.

References

Bø, I.(1978), Ungdom og naboland. En undersøkelse av skolens og fjernsynets betydning for nabospråksforståelsen, *Technical Report 4*, Stavanger.

Braunmüller, K.(2002), Semicommunication and accommodation: Observations from the linguistic situation in Scandinavia, *International Journal of Applied Linguistics*.

Delsing, L. and Lundin Åkesson, K.(2005), *Håller språket ihop Norden?*, Nordiska ministerrådet, Copenhagen, Denmark.

Edlund, L. and Hene, B.(1992), *Lånord i svenskan - om språkförändringar i tid och rum*, Förlags AB Wiken, Umeå and Stockholm, Sweden.

Gooskens, C.(2006), Linguistic and extra-linguistic predictors of Inter-Scandinavian intelligibility, *in* J. van de Weijer and B. Los (eds), *Linguistics in the Netherlands*, Vol. 24, John Benjamins, Amsterdam, pp. 101–113.

Gooskens, C., van Bezooijen, R. and Kürschner, S.(submitted, 2007), The lexical profiles of Dutch and Swedish. A contrastive study, *in* B. Los and M. van Koppen (eds), *Linguistics in the Netherlands*, Vol. 23, John Benjamins, Amsterdam.

Haugen, E.(1966), Semicommunication: the language gap in Scandinavia, *Sociological Inquiry* **36**, 280–297.

Heeringa, W.(2004), *Measuring Dialect Pronunciation Differences using Levenshtein Distance*, PhD thesis, University of Groningen.

Maurud, Ø.(1976), Reciprocal comprehension of neighbour languages in Scandinavia, *Scandinavian Journal of Educational Research* **20**, 46–52.

Omdal, H.(1995), Attitudes toward spoken and written Norwegian, *International Journal of the Sociology of Language* **115**, 85–106.

5

Which New York, which Monday?

The role of background knowledge and intended audience in automatic disambiguation of spatiotemporal expressions

Ineke Schuurman
Katholieke Universiteit Leuven

Abstract

The aim of MiniSTEx, a system for automatic spatiotemporal annotation, is to locate eventualities on a time-axis and to disambiguate geospatial information in such a way that geospatial entities can be located on a map. Therefore all kinds of spatiotemporal (geospatial, temporal and geotemporal) expressions are disambiguated. In doing so, the concepts of "background knowledge" and "intended audience", together with the Gricean maxims, play an important role, especially when dealing with indexicals. The system relies on a database containing all kinds of spatiotemporal expressions. At the moment MiniSTEx is used for both Dutch and English texts.

5.1 Introduction

MiniSTEx is a first version of a larger annotation system for spatiotemporal phenomena under construction.[1] It has to handle all types of Dutch texts (both fiction

[1]I would like to thank my colleagues, and especially Vincent Vandeghinste, for all discussions.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

and non-fiction, i.e. novels, newspapers, web pages, pamphlets, etc.). The general spatial part of the system still needs to be developed in more detail in the future. The geospatial part, however, is already handled. The same holds for the temporal and geotemporal parts.

The aim of this annotation scheme is to identify spatiotemporal expressions, and to normalize and disambiguate them in order to facilitate reasoning. The approach is meant to be used in applications like (multi-lingual) information retrieval, question answering, and multidocument summarization.

MiniSTEx reflects the state of the art in geospatial and temporal annotation. With respect to the latter, TimeML (Sauri et al. 2006) and TIDES (Ferro et al. 2005) come to mind. Geospatial annotation as such is far less widespread and standardized. However, the subtask of disambiguation is also a subject in geographic information extraction. Some approaches in this field can be found in Ding et al. (2000), Leidner (2006), and Volz et al. (2007).[2]

Typical for MiniSTEx is that it handles a) both geospatial and temporal expressions, and b) also geotemporal expressions, i.e. expressions associated with a combination of geospatial and temporal properties. The system was designed to be used in circumstances in which the background of the texts is known, i.e. not in the first place for web pages and the like. In the annotation process a large spatiotemporal database plays a central role.
And pragmatics, especially when using both the background of a text and its intended audience, plays an important role in deciding which database entry is to be associated with a particular spatiotemporal expression in a text when the tokens as such can refer to several concepts: "which New York, which Monday?"

In the STEVIN-project[3] SONAR (2007-2010) a syntactically analyzed subcorpus[4] of Dutch is being enriched with four types of semantic annotation: a) named entity identification and classification, b) coreference resolution, c) semantic roles and d) spatiotemporal relations (the latter using MiniSTEx). Within SONAR at least part of the expressions to be identified and disambiguated (the so-called timexes) by MiniSTEx are already marked as such.
MiniSTEx is also used in the SBO-project AMASS++ (Advanced Multimedia Alignment and Structured Summarization), funded by IWT. In AMASS++ (2007-2011) we use it both for Dutch and English.
In this paper we will pay special attention to the strategy used to select referents for contextually dependent, non-deictic expressions.

[2]Note that we annotate more phenomena than covered in these papers, cf. Schuurman (2007).
[3]`http://stevin-tst.org`.
[4]SONAR is a 500-million-word reference corpus of contemporary written Dutch. A 1 million subcorpus will be semantically annotated.

5.2 Which New York, which Monday?

There are over 50 Mondays in a year, and, according to Wikipedia[5] (English version, March 2007), 8 geospatial entities called New York, cf. table 5.1.

Table 5.1: Which New York?

New York	U.S. state (population
New York	city in the above state
New York	county, generally referred to as Manhattan
New York	metropolitan area
New York	Lincolnshire
New York	Tyne and Wear
New York	Missouri
New York	Texas

Even as a geospatial expression *New York*[6] is ambiguous. Even more ambiguous than shown in table 5.1: in GeoNet Names Server[7] (a gazetteer) there are already 12 hits outside the US. And in the Getty Thesaurus of Geographic Names[8] 15 instances inside the US are mentioned. In our database-driven approach this means that an expression like *New York* might get several entries in the spatiotemporal database (up to 27+).

So which one to choose when annotating a particular text?

One of the basic assumptions of MiniSTEx is that in order to facilitate reasoning quantification of information is essential. Therefore, in contrast with common practice, cf. Sauri et al. (2006), expressions like *winter* are also normalized in terms of the months people associate with *winter*, for example *december, january* and *february*: "XXXX-12/02"[9] (instead of "XXXX-WI").[10]

Note that in Schuurman (2007) some spatiotemporal expressions may have various, in se correct values, depending for example on the hemisphere (*winter*), or on religion and/or tradition (*Christmas*). Others are often used in a sloppy way, like *winter, week* or *Christmas*[11]. Reliability features (`noise` and `soft`) are added to indicate such behaviour, cf. Schuurman and Monachesi (2006), and especially Schuurman (2007), when it is not clear enough which referent is meant.

People do succeed in detecting the correct referent from context. MiniSTEx is able to do so as well, i.e. to identify spatiotemporal expressions, and to disambiguate them.

Before we describe the MiniSTEx approach, let us have a look at the kind of spatiotemporal data we typically are confronted with when annotating (or reading)

[5] `http://en.wikipedia.org`.

[6] There are also lots and lots of hotels, ships, songs, albums, etc with this name. Within the Stevin-programme named entitiy recognition is to filter out these.

[7] `http://earth-info.nga.mil/gns/html/index.html`.

[8] `http://www.getty.edu/research/conducting_research/vocabularies/tgn/`.

[9] In combination with a 'reliability' feature when necessary, cf. Schuurman (2007).

[10] In which 'WI' is an abbreviation of *winter*.

[11] For example: when you are going somewhere for Christmas, is it just the 25th of December, or does it include the 26th as well?

a text in order to detect anchors enabling the location of eventualities (events, states, processes) on a time-axis and/or on a virtual map.

5.3 Types of times and places in need of disambiguation

Whereas the expressions in section 5.3.1 contain all information needed to interpret them themselves, this is not the case for those in section 5.3.2.

5.3.1 Independent temporal, resp. geospatial expressions

Examples of independent temporal, resp. geospatial expressions are expressions like those in a) *March 1st, 2003; Washington D.C.; the Netherlands* and b) *the first Tuesday in May 2000, the capital of Sweden*.
Of the expressions mentioned only those in a) are really easy to describe formally. They (or their constituting elements, as for *March 1st, 2003*) are contained as such in the database, cf. table 5.5.[12]

In expressions like *the first Tuesday in May 2000* or *the capital of Sweden*[13] the constituting elements need to be solved before a specific date or town can be associated with them. For *the first Tuesday in May 2000*, etc. this means that the constituting elements are contained in the database as forms, to be combined and solved when applied: "2000-05-02".
The common characteristic of all these expressions is that there is just one possible solution, even when part of the construction can refer to several temporal or geospatial entities.

5.3.2 Indexicals

Indexicals are context dependent expressions, usually deictic ones, like *today, this week, now; here, in this country*.
But note that also the meaning of *March 1st, Monday, Easter 2003, winter 2002* and *New York, Dallas* and *Washington* depends on the broader context or even other information coming with the text under consideration (metadata). In the first two expressions the year is lacking, *Easter* comes on another date in the orthodox church, *winter* depends on the hemisphere, *New York* can be the city or the state, etc.

Such indexicals need to be solved, taking the context into consideration. This not only is necessary for deictic expressions, but is also explicitly necessary for non-deictic expressions like *Monday* or *New York*: which *Monday*, which *New York*?

It are expressions like these non-deictic ones that are the subject of this paper.

[12] The database as presented here is a simplified one.
[13] In order to solve this construction, we need an additional (optional) feature in the `geo`-tag of *Stockholm*, expressing that it is a capital.

5.4 Indexicals and the interpretation thereof

The problem of how to annotate *Monday*, *New York*, and the like mainly concerns the interpretation of both temporal and geospatial indexicals, cf. section 5.3.2.

Table 5.2: Which one to choose?

Groningen	province or town in the Netherlands
Den Haag,'s Gravenhage	several names for the same town
Vecht	2 rivers in the Netherlands
Rijn	same river in several countries
Luxemburg	country, town in that country, or province in Belgium
Haren	2 villages in the Netherlands, one in Belgium
Kerst (Christmas)	on different dates depending on religion
vaderdag (father's day)	many dates possible, depending on country/region
winter	different months, depending on hemisphere; different dates (meteorological vs astronomical winter)
Koninginnedag	April 30 since 1949; August 31st from 1890 till 1949
november revolutie (November revolution)	same as October revolution
namiddag (afternoon)	different periods of time in the Netherlands and Belgium
Rio de Janeiro	town, region or Earth Summit[14]

A look at table 5.2 shows us that there is a variety of cases to disambiguate. The examples all show different instances of what in (geographical) information extraction is called[15]

1. *multi-referent ambiguity* (or homonymy): two or more concepts share the same name (*Groningen, Haren, hofstad, vaderdag, winter*)
2. *name-variant ambiguity* (or synonymy): the same concept comes with several names (*Den Haag – 's Gravenhage, november revolution – october revolution*[16])

Whereas we are not aware of attempts to solve problems like these as far as temporal concepts are concerned, there are a few attempts with respect to geospatial concepts in the field of information extraction, cf. section 5.1.

5.4.1 Other (geospatial) approaches

In Volz et al. (2007) a novel approach is presented to disambiguate geographic names based on an ontology. Their ontology contains data from publicly available

[14]This is an example of a geotemporal expression. Whereas geospatial expressions are in fact a subset of spatial expressions, geotemporal expressions are expressions associated with both temporal and geospatial properties. These are typically expressions concerning larger events like (*fall of the Berlin wall, Earth Summit, 9/11, . . .*), the temporal and/or geotemporal details of some of these may even be considered common knowledge (*World War II*). Albeit sometimes there is some uncertainty whether it started in 1939 or 1940, or about the exact end date, the intended audience knows where to situate World War II on a time axis.

[15]We will bypass the third ambiguity: geoname (short for geographical name) – non-geoname.

[16]Depending on the calendar used.

gazetteers (like GeoNet Names Server) and common world/linguistic knowledge obtained from WordNet[17] and EuroWordnet.[18] When they have spotted all candidates for geonames, they first try to narrow down the selection by looking in a window of 2 consecutive geographical terms whether there are clues to be found (like *Paris, France* vs. *Paris, Texas*), in a second step a window of 11 consecutive terms (t_i(+|-)5) is taken into consideration to find instances of the same geographic feature class (like *country, populated place*).
The remaining candidates are ranked according to the weights attached to the concepts in the ontology. A country gets the weight +3000, a populated place the same weight (+3000), but in this case the number of inhabitants (divided by 1000) is added. This would mean that, when no further information is availble via the first steps, the city of Luxemburg will be ranked higer than the country with the same name, and that Lancaster (California) will be ranked higer than Lancaster (UK).

Ding et al. (2000) are closest to our approach in that they try to determine the intended audience of a webpage, i.e. its geographical scope. They use two methods to determine this scope: 1) look what the geographic location is of hosts referring to the website under consideration, and 2) look what the scope is of all geographical places mentioned in this website. This will give a clue where the intended audience is located.

5.4.2 The pragmatic MiniSTEx approach

In order to develop a system dealing with disambiguation of temporal and geospatial data we asked ourselves "What makes a reader understand the geospatial and temporal data contained in a specific text?" as such characteristics may be useful for our design as well.

The vital property of a text seems having an intended audience: a medical text written for British GPs is likely not to be fully understandable for either aerospace engineers, teachers or linguists. Nor for Belgian GPs. And a text written for people living in Amsterdam, be it a newspaper or a bulletin by the city council, may not be understandable for people living in Brussels or Rotterdam when refering to local information.

This is the case because every speaker (author) will apply conversational maxims as formulated by Grice (1975), often paraphrased as "Don't say too much and don't say too little.", cf. Dale and Reiter (1996), without as much as thinking.
These maxims are

1. Maxim of Quantity:
   (a) Make your contribution as informative as required;
   (b) Do not make your contribution more informative than is required.
2. Maxim of Relation (or Relevance):

[17] `http://wordnet.princeton.edu/w3wn.html`.
[18] `http://www.illc.uva.nl/EuroWordNet/`.

(a) Be relevant.

3. Maxim of Manner:

   (a) Be perspicious:

      i. avoid obscurity of expression,
      ii. avoid ambiguity,
      iii. avoid unnecessary wordiness,
      iv. be orderly.

4. Maxim of Quality:

   (a) Do not say what you believe to be false;

   (b) Do not say that for which you lack evidence.

In MiniSTEx, we assume that a text always provides the (intended) reader with all information necessary to understand this text. If not, i.e. when a human reader belonging to the intended audience fails to understand a text, a system can neither be blamed for failing. MiniSTEx handles texts by using the background and world knowledge the intended audience is supposed to have.
Therefore problems we are faced with are:

(A) Determination of the intended audience of a text

(B) Determination of the corresponding spatiotemporal background knowledge

(C) Exploitation of this background knowledge

5.5 Determination of intended audience and spatiotemporal background knowledge

As far as problem (A) is concerned, note that our approach is not designed to primarily deal wit web pages, but rather with digital archives (broadcasting companies, news agencies), corpora and the like. Of the latter kind of resources the background is more often known. This is very important as it helps us a lot in determining both (A) the intended audience and (B) the spatiotemporal background knowledge this audience may be supposed to possess. So, unlike Ding et al. (2000) working with web pages in English, we do not solely rely on the distribution of web links in determining the intended audience. A first clue is provided by the language used: a text written in Dutch is in all probability meant for Dutch and/or Flemish readers, a text in Hebrew for Israelis or Jews around the world. For texts in English, the intended audience is more difficult to discover as these are either meant for a British (or an American, Australian, Canadian,...) audience, i.e. the text has a national scope, or for "the rest of the world" (global scope). But, especially for the smaller languages, data with respect to the intended audience can thus be derived even when details with respect to the source of the text are unknown. However, for known resources many more details are available,

making use of the spatiotemporal data associated with the title (like *De Morgen*, *Daily Telegraph*, *Boston Globe*, *www.vlaanderen.be* etc)., cf. table 5.3.[19]

Table 5.3: Background-doc

concept	dbid	status	geo	trad	cal	lang	scope
De Morgen	220000	newspaper	Brussel			Dutch	national
De Telegraaf	220003	newspaper	Amsterdam				national
Ref. Dagblad	220009	newspaper	Apeldoorn	orth-ref			
Vl.overheid	230000	web	Brussel			Dutch	regional
Vl.overheid	230000	web	Brussel			English	global

Other information relevant for determining the intended audience are *tradition* (Christian, Islamic, Jewish, Eastern Orthodox, ...), and *calendar*: (Gregorian, Hebrew, Hindu, ...). Nowadays the Gregorian calendar is widely used in Israel, but the Hebrew calendar can also still be used (and is in fact used in a religious or cultural context). And the November Revolution in Russia (1917-11-07 according to the Julian Calendar used in Russia at that moment) is known in the western world as the October Revolution (Gregorian calendar: 1917-10-25). The intended audience of a Jewish newspaper or an older Russian text is supposed to be familiar with such traditions and calendars.

Table 5.4: Background-geo

concept	dbid	status	trad	cal	hem	UTC[20]	lang	partof	division
Spanje	109	cntry	chr	Greg	north	+1	ES	EU	2=region, 3=province
Nederland	146	cntry	chr	Greg	north	+1	DU	EU	2=–, 3=province
België	137	cntry	chr	Greg	north	+1	DU, FR, GE	EU	2=region, 3=province
VS	199	cntry	chr	Greg	north	-(5/10)	EN, ES	NA	2=state, 3=county
Vlaanderen	102	region					DU	BE	

The MiniSTEx database consists of more tables than presented in this section, cf. the tables in section 5.5.1. Those tables provide the data to connect the concepts in these background tables: in table 5.3 the geo-column refers to geospatial entities. Via table 5.5 these entities can be linked with entities in table 5.4. This table defines the spatiotemporal backgroundknowledge associated with a geospatial entity, unless it is superseded by information in table 5.3 itself. These columns in table 5.3 are only filled out in case they contain information that is to overrule the general information. So, *Reformatorisch Dagblad* is said to belong to the

[19] For convenience of the reader most tables as they are presented here contain the concepts. This is only for matter of presentation. In reality the only column all tables contain is the one with the `dbid`. The real tables also contain more columns, i.e. more types of data. And there are more tables.

[20] Coordinated Universal Time.

orthodox-reformatoric tradition instead of the more general christian tradition. For *De Morgen* and *De Telegraaf* the values for `geo` and `trad` are those of *Brussel* and *Amsterdam* respectively. For *De Telegraaf* `lang` is also that of *Amsterdam*, whereas for *De Morgen* the values for *Brussel* are overruled by the statement that only *Dutch* is used.

In MiniSTEx the spatiotemporal background knowledge the intended audience is supposed to have is contained in a series of tables.

5.5.1 The design of the MiniSTEx spatiotemporal database

As might be expected from the previous sections, the MiniSTEx database is meant to mimick the spatiotemporal knowledge of an intended audience. It is not the case that a new database is built for every new audience (one for *De Morgen*, another one for texts by the *Vlaamse overheid*, still another one for *Reformatorisch Dagblad*, etc.). This is not necessary, although parts of the database, like `ranking`, will need to be adapted for other 'supertypes' of intended audience (other countries etc). This issue will be researched in AMASS++.
In the end the Dutch database, consisting of a series of tables. will contain lots of temporal and (geo)spatial data with respect to the Dutch language, the Netherlands and Belgium, but far less with respect to, say French Guyana, Peru and Macedonia, the jewish calendar and the orthodox culture. Of these it will contain only those data relevant for a Flemish/Dutch audience. It may, for example, only contain two instantiations of New York (the state and the metropole).[21] This makes our approach a pragmatic one.

The central table in our database, cf. table 5.5, contains the concepts, their `dbid` and the `tag` associated with them, together with their `background`, `rank` and `parts`.
The `background` of a concept refers to specific conditions associated with it. *Thanksgiving* is celebrated both in the USA and in Canada, but on different dates. Apart from such geospatial conditions, references may be made to `tradition`, `calendar`, `hemisphere`, `language` (this one albeit rather seldom), ...[22] It might come as a surprise that language doesn't play a more important role. But it turns out that the role of the country, or the region, is by far more important. The case of *vaderdag* is illustrative in this respect. At least three values for *vaderdag* are valid in the Dutch-speaking regions, cf. table 5.5. But when an item on, say, the Antwerp *vaderdag* is translated into English the term used will become *Father's day*, although the date it refers to is still to be the Antwerp one, not the UK one! The geographical background is of importance, not the language used.

The ranking indicates that, when *Thanksgiving* is mentioned in a Dutch or Flemish context without further details, it is likely to refer to the American instantiation (see below).

As alluded to above, our database for Dutch contains many (corrected) data relevant for the Netherlands and Belgium, based on gazetteers, Wikipedia,

[21] Others to be added when need arises.
[22] Cf. below, the paragraph on background.

(Euro)WordNet, etc. New data are added constantly. For other countries it contains only those data we consider relevant (like all continents, all countries, main cities in the neighbouring countries, and the US, main rivers etc.), based on the same kind of resources, plus some others, like The World Factbook[23]. More will be added when necessary on basis of the texts handled by the system. Therefore it is likely that for *New York*, cf. table 5.1, only the top two will ever be contained in it.[24] This means that names that in se could be ambiguous according to a gazetteer or Wikipedia can be unambiguous in our (Dutch) database.

A second table, cf. table 5.6, contains the name variants of the concepts contained in table 5.5, like synonyms. But still only in Dutch. Here again we use ranking to indicate the most likely referent(s).

There also is a, rather small, table with language-sensitive concepts, cf. table 5.7. Above we have explained that in general all and every of the background factors is of greater importance than the language[32]. There are just a few exceptions, in which a language only allows one value to be associated with a concept, while in other languages these concepts are associated with other values. An example that comes to mind is *avond – evening* vs *nacht – night*.

Table 5.7: Language-sensitive concepts

concept	dbid	language	tag
avond (evening)	1302	Dutch	<temp type="cal" val="T18/24">
nacht (night)	1303	Dutch	<temp type="cal" val="T22/06">
evening	1308	English	<temp type="cal" val="T18/21">
night	1309	English	<temp type="cal" val="T21/06">

Although the values given for many concepts may vary to some extent from person to person, from region to region, or from season to season, the ranges are relatively small, i.e. it is a matter of `noise`, not from a completely different value for a different, albeit related, concept. But in case of *avond – evening* vs *nacht – night* the Dutch and British concepts are different ones (reflected by the `dbid` they get).

The first step in selecting referents is to determine all non-ambiguous expressions. On basis of these the value of the remaining expressions is calculated in the next steps, keeping in mind the background of the text, cf. tables 5.3 and 5.4, and the type of the surrounding names: when *Kerst* (Christmas) appears in a text with a background in the christian tradition, it will be solved as referring to the

[23] https://www.cia.gov/library/publications/the-world-factbook/geos/be.html.
[24] *Manhattan*, for example, is unlikely to be referred to as *New York*, although it will be linked.
[25] The '::'-sign is only used in geospatial entities. A::B indicates that B is part of A.
[26] 'greg' refers to the Gregorian calendar
[27] 'form' is used instead of 'val' when variables are involved (in this case for the year, indicated by XXXX).
[28] A '|' is used to indicate a non-exclusive '*or*', the brackets indicate the scope.
[29] 'X..Y' indicates one or more elements out of a range X till Y.
[30] 'X/Y' means the whole range X up to and including Y
[31] An example of a geotemporal concept, thus including other concepts.
[32] Although the language is important in determining the intended audience.

Table 5.5: Concepts

concept	dbid	background	tag	rank	parts
Spanje	109	EU[25]	<geo type="country" val="EU::Spanje"/>		
Brussel	130	BE::BR	<geo type="place" val="EU::BE::BR::-::Brussel" />		
Den Haag	135	NL::ZH	<geo type="place" val="EU::NL::-::ZH::Den Haag" />		
Apeldoorn	145	NL::GE	<geo type="place" val="EU::NL::-::GE::Apeldoorn" />		
Haren	142	BE::BR	<geo type="place" val="EU::BE::BR::-::Haren />	2	
Haren	143	NL::GR	<geo type="place" val="EU::NL::-::GR::Haren />	1	
Haren	144	NL::NB	<geo type="place" val="EU::NL::-::NB::Haren />	3	
augustus	10057	greg[26]	<temp type="cal" form="XXXX-08"[27]/>		
vaderdag	1500	EU::(NL|UK|FR)[28]	<temp type="cal" form="XXXX-06-D07,15..21"[29]/>		
vaderdag	1501	EU::BE	<temp type="cal" form="XXXX-06-D07,08..14" />		
vaderdag	1502	BE::AN	<temp type="cal" form="XXXX-03-19" />		
St. Jozef	1550	chr	<temp type="cal" form="XXXX-03-19" />		
Thanksgiving	210074	NA::VS	<temp type="cal" form="XXXX-11-D04,22..28" />	1	
Thanksgiving	210075	NA::CA	<temp type="cal" form="XXXX-10-D01,08..14" />	2	
avond	1302	DU	<temp type="clock" form="T18/24" />		
nacht	1303	DU	<temp type="clock" form="T22/06" />		
middag	1291	EU::NL	<temp type="clock" val="T12/18"[30]		
namiddag	1292	EU::NL	<temp type="clock" val="T16/18"		
namiddag	1293	EU::BE	<temp type="clock" val="T12/18"		
Kerst	1310	chr	<temp type="cal" form="XXXX-12-25" />		
Kerst	1311	orth	<temp type="cal" form="XXXX–01-07" />		
winter	100562	north	<temp type="cal" form="XXXX-12/02" />		
Rio de Janeiro	101	BR::RJ	<geo type="place" val="SA::BR::RJ::-::Rio de Janeiro" />	1	
Rio de Janeiro	141	SA::BR	<geo type="region" val="SA::BR::Rio de Janeiro" />	2	
UNCED[31]	500010	UN|conf	<stex> <temp type="cal" val="1992-06-3/14" /> </stex>		101

Table 5.6: Name-variants of concepts

name-variant	dbid	concept	rank
's Gravenhage	135	Den Haag	
hofstad	135	Den Haag	1
hofstad	145	Apeldoorn	2
oogstmaand	10057	augustus	
Rio-conferentie	500010	UNCED	
Rio	500010	UNCED	
Rio de Janeiro	500010	UNCED	
VN-conferentie inzake ontwikkeling en milieu	500010	UNCED	
wereldmilieu- en ontwikkelingsconferentie	500010	UNCED	

25th of December, unless it refers to *Kerst* in for example Russia. Russia comes with an orthodox background, and therefore *Kerst* will be solved as occuring in January, which is to be indicated. *Haren* will be associated with the village in the Brussels Capital Region when mentioned in an item in De Morgen (unless stated otherwise) because of its background.

5.6 The disambiguation steps

One could say that according to the Gricean maxims the intended audience of a text, cf. problem (A), in fact determines the way the content of a text is articulated. It is the intended audience, together with its (spatiotemporal) background, that makes an author mention things explicitly, or leave them out.
For example, in Belgium everybody knows that the official languages of the country are Dutch, French and German; or that Leuven is a town in Flanders, one of the three regions in Belgium. This is not mentioned in a text for, say, a Flemish audience. Indeed, a text becomes almost unreadable when it contains such unnecessary details. But in a British newspaper, one will have to mention such Belgian details explicitly. Another example: for a Flemish audience it is obvious that *Sinterklaas*[33] is celebrated on the 6th of december, whereas in the Netherlands the 5th will be associated with it. So, when in a Dutch newspaper an item would occur on the celebration of *Sinterklaas* in Belgium, the date will be mentioned explicitly. Otherwise, the intended Dutch readers will assume it is the 5th, as they are used to.

As a consequence, the intended reader by default prefers one reading over another one as a consequence of his spatiotemporal background knowledge, and the expectations evoked by it. In the same way the reader expects the referent of a geospatial expression to be the one that is most relevant for him (for example the most nearby *Haren* or the most well-known *New York*), he also expects temporal referents to be as relevant as possible. A plain reference to *Monday* is taken to

[33]The name day of Saint Nicholas, patron saint of, among others, children.

refer to last Monday or next Monday[34]; a reference to Christmas to the dates he himself is used to celebrate it. When another instantiation is desired, this should have been made clear.

The last refuge for both human reader and MiniSTEx is ranking: when all other steps fail one has to look at the importance of a specific referent for a intended audience. Clues may be nearness, importance, size, ...
Unlike Volz et al. (2007), we do not rank towns always higher than countries, or countries always higher than provinces, or a larger town higher than a smaller one. Understandably, this is the only way to attach ranks automatically for all geographic names in a gazetteer.
As we use just a selection, adding new names one by one, we can afford to attach ranks another way.

When the intended audience of texts published in Western Europe is confronted with a geographic name *Dover*, they are inclined to associate it with the town of that name in the United Kingdom, and not with the capital of the state of Delaware, USA, although the latter is the larger one. Also, the country of *Luxemburg* has a better ranking than its capital with the same name, and the same with respect to the Walloon province with that name, even in a Belgian text. A rule of thumb is that referents in neighbouring countries are preferred over referents in further away countries. Also the relations between the countries involved may play a role (export relations, former colonies, etc.).

The highest rank available is `1`, and several concepts sharing the same name may occasionally have the same (low) rank. In such a case two or more referents will be provided as value. The same holds mutatis mutandis for temporal referents.

In MiniSTEx, the steps taken for disambiguation are roughly the following (after each step elements are disambiguated (if possible), unless the results are contradictionary, in which case the next step is applied):

1. identification of unambiguous spatiotemporal elements;
2. identification of all general spatiotemporal (broad sense) expressions (*stad* (town), *land* (country), *noordelijk halfrond* (northern hemisphere), *christelijk* (christian), *burgemeester* (mayor), ...;
3. confrontation of these with ambiguous referents, first at the level of the constituent (*de stad Antwerpen*) (the city of Antwerp)), later at that of the sentence and the paragraph;
4. selection of readings coming with the same `background`, cf. section 5.6.2;
5. identification of the `division` value, cf. table 5.4, of the referents solved unambiguously;
6. select the reading with the best rank.

[34]The choice between 'last' or 'next' is guided by the tense of the verb. But note that a plain *Monday* never will be taken to refer to a Monday several weeks ago or in the future. It has to be `last` Monday.

5.6.1 One sense per text

When looking at a set of names (in a list, or in a window of *x* terms) the types and values of the unambiguous names will steer the interpretation of the ambiguous ones:

(1) {Antwerpen,Leuven,Utrecht,Groningen}

(2) {Antwerpen,Vlaams-Brabant,Utrecht,Groningen}

Just by the fact that *Leuven* is a town, whereas *Vlaams-Brabant* is a province, the other three names in the set are towns, resp. provinces as well, although they are not in the same country.

5.6.2 Additional world knowledge

The intended audience is expected to have some additional world knowledge.[35] Not only in order to be able to handle concepts like *World War II* when these occur without further details, but also for disambiguation purposes. The reader should for example be able to deduce the correct referent for *Antwerpen* (town or province) in expressions like the following:

(3) de burgemeester van Antwerpen
the mayor of Antwerp

(4) de gouverneur van Antwerpen
the governor of Antwerp

In (3) the town of *Antwerpen* is meant, as a province does not have a mayor, whereas a town does, and in (4) just the other way round.

5.7 Conclusion

For automatic spatiotemporal annotation, and especially disambiguation, of a text it turns out to be important to know the intended audience of that specific text. One needs to know *when*, *where* and *in which context* (which newspaper, website, ...) a text appeared. That way, the spatiotemporal knowledge a reader (and a system) needs in order to understand the text can be derived.
Of course, one does not always have all these details. But, except for English and other global languages (like French and Spanish), the language used already gives a clue. Furthermore, the geographical scope as used by Ding et al. (2000) will provide some details, also for English texts. But in case more data are available, one should consider using them.
We just started adding data for English, as the system was originally designed for Dutch[36] only. Up till now we are using one large database for both languages. Whether it is to be split is still researched.

[35] Relevant for the specific part of the world under consideration.
[36] Both the variants as spoken in the Netherlands and Flanders.

In the future general spatial concepts will be added, which are more complex than the geospatial ones.

References

Dale, R. and Reiter, E.(1996), The role of the Gricean maxims in the generation of referring expressions, *in* B. D. Eugenio and N. L. Green (eds), *Working Notes: AAAI Spring Symposium on Computational Implicature: Computational Approaches to Interpreting and Generating Conversational Implicature*, American Association for Artificial Intelligence, Menlo Park, California, pp. 16–20.

Ding, J., Gravano, L. and Shivakumar, N.(2000), Computing geographical scopes of web resources, *26th International Conference on Very Large Databases, VLDB 2000*, Cairo, Egypt.

Ferro, L., Gerber, L., Mani, I., Sundheim, B. and Wilson, G.(2005), *TIDES 2005 Standard for the Annotation of Temporal Expressions*.

Grice, H.(1975), Logic and conversation, *in* P. Cole and J. Morgan (eds), *Speech Acts*, Vol. 3 of *Syntax and Semantic*, Academic Press, New York, pp. 43–58.

Leidner, J.(2006), Toponym Resolution: A First Large-Scale Comparative Evaluation, *Technical report*, School of Informatics, University of Edinburgh.

Sauri, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A. and Pustejovsky, J.(2006), *TimeML Annotation Guidelines, version 1.2.1*.

Schuurman, I.(2007), *MiniSTEx Protocol, version 0.2*, Centre of Computational Linguistics, K.U.Leuven. KULeuven 2007.

Schuurman, I. and Monachesi, P.(2006), The contours of a semantic annotation scheme for Dutch, *Proceedings of CLIN 2005*.

Volz, R., Kleb, J. and Mueller, W.(2007), Towards ontology-based disambiguation of geographical identifiers, *WWW2007*, Banff, Canada.

6

Discovery of association rules between syntactic variables

Data mining the Syntactic Atlas of the Dutch Dialects

Marco René Spruit
Meertens Instituut

Abstract

This research applies an association rule mining technique to purely syntactic dialect data. The paper answers the research question of how relevant associations between syntactic variables can be discovered. The method calculates the proportional overlap between geographical distributions of syntactic microvariables and incorporates rule quality factors such as accuracy, coverage and completeness to measure the interestingness of the variable associations. The exploratory review of the results discusses several highly ranked association rules and also examines an implicational chain of syntactic variables.

6.1 Introduction

This work[1] investigates a data mining technique to discover associations between syntactic variables in Dutch dialects using a rule induction system based on proportional overlap. The research aims to contribute to the understanding of the as-

[1]The research for this paper is being carried out in the context of the NWO project The Determinants of Dialectal Variation, number 360-70-120, P.I. J. Nerbonne. Please visit http://dialectometry.net for more information and relevant software.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

sociations between syntactic variables by examining geographical distributions of syntactic microvariation. The current paper addresses the following two research questions:

1. How can relevant associations between syntactic variables be discovered?
2. What are interesting associations between syntactic variables?

This research integrates expertise from the research fields of data mining and ecology to answer these questions quantitatively. In essence this investigation exhaustively evaluates levels of association between combinations of syntactic variables based on the proportional overlap between their geographical distributions.

This work proceeds from the observation that linguistic research frameworks such as generative syntax and functional typology share a primary interest in understanding the structural similarities and differences between language varieties. The frameworks aim to identify which universal syntactic properties can vary across language varieties and which remain constant. The ultimate goal is to characterise the superficial structural diversity of all language varieties as particular settings of relatively few parametric patterns. Unfortunately, the search for syntactic universals is still very much a topic of ongoing research. Gianollo et al. (to appear) most notably define an extensive parametric framework to model language variation in the internal structure of Determiner Phrases based on a relatively wide sample of languages and language families.

Haspelmath (to appear) compiles a list of seven universal syntactic parameters for which there is a wide consensus in the field. One well-known example of a syntactic universal is the pro-drop/null-subject parameter, which states that the subject position in a clause may be empty or must be filled by a subject pronoun. It was originally thought to universally correlate with syntactic phenomena such as null thematic subjects and null expletives (Rizzi 1986). However, the generalisation quickly became untenable once more language varieties were analysed (Newmeyer 2005). This example adequately illustrates that a large data set of comparable language varieties is required to investigate syntactic variable relationships more reliably. Such an examination needs to be automated using verifiable methods because of the exhaustive and repetitive nature of the comparison procedure.

The current research aims to contribute to the global research effort of parametrisation of the structural diversity of language varieties by proposing a computational method to discover syntactic variable associations automatically. The technique facilitates exploration of previously unknown variable relationships and validation of existing parametric generalisations. The second research question is addressed through an exploratory review of the method's application to a large syntactic microvariation database.

The paper is structured as follows. Section 2 describes the unique syntactic variation database under investigation. Section 3 introduces the sample data subset used in Section 4 to illustrate the association rule mining procedure based on proportional overlap. Section 5 reviews the evaluation factors to accurately measure the quality of the association rules. Section 6 explores the most interesting rules

discovered in the sample data. Section 7 highlights results of the association rule mining application to the entire syntactic variation database under investigation. Section 8 recapitulates the main findings. The paper concludes with a discussion and directions for future research in Section 9.

6.2 Syntactic variation database

This research examines the first volume of the *Syntactische Atlas van de Nederlandse Dialecten* (SAND1; 'Syntactic Atlas of the Dutch Dialects'; Barbiers et al. (2005)) from a quantitative perspective. SAND1 contains 145 geographical distribution maps of individual syntactic variables in 267 Dutch dialects in the Netherlands, the Northern part of Belgium and a small northwestern part of France.[2] It covers syntactic variation related to the left periphery of the clause and pronominal reference. This includes variation with respect to complementisers, subject pronouns and expletives, subject doubling and subject clitisation following yes/no, reflexive and reciprocal pronouns, and fronting phenomena. The second and final volume of the SAND is due to appear near the end of 2007 and will describe syntactic variation in Dutch dialects with respect to verbal clusters, negation and quantification. Cornips and Jongenburger (2001) review the methodological aspects of the written and oral syntactic elicitation techniques which were employed to reliably collect the SAND data.

From a quantitative research perspective SAND1 also represents a syntactic microvariation database containing 106 syntactic contexts and 485 syntactic variables among varieties of a single language. This work defines a syntactic variable as a form or word order in a syntactic context in which two dialects can differ (Spruit 2006). The number of available syntactic contexts is somewhat lower than the number of geographical maps because SAND1 also contains numerous correlation maps which show syntactic variables from different perspectives. Also, some syntactic contexts are presented using multiple maps.

Tables 1 to 4 provide examples of syntactic variation in the complementisers, subject doubling, reflexives and fronting domains, respectively. For example, Table 1 shows the attested variation throughout the Dutch language area in the realisation of the complementiser position in comparative if-clauses as presented in SAND1 map B on page 14. In standard Dutch people say *'t lijkt wel of er iemand in de tuin staat* 'it looks [affirmative] if there someone in the garden stands', but in colloquial Dutch the following form also frequently occurs in the southern provinces: *'t lijkt wel of dat er iemand in de tuin staat*. There are even a few northern and southern regions within the Dutch language area where the verb occurs in the second position of the if-clause: *'t lijkt wel of er staat iemand in de tuin*. The last example also illustrates that both word form and word order may vary within a syntactic context.

[2]The online version of this paper at http://marco.info/pro/pub/mrs2007clin.pdf includes geographical distribution maps of the SAND dialect locations with relevant province names and also contains additional data mining results and references.

Table 1. Map 14b in SAND1 shows seven syntactic variables in the complementisers domain.

Context	Complementiser of comparative if-clause											
Variables	{ of, *of dat, dat, as/of + V2, at, as, et }											
Example	't	lijkt	wel	of	dat	er	∅	iemand	in	de	tuin	staat.
Gloss	it	looks	[affirmative]	if	that	there	[v2]	someone	in	the	garden	stands
Translation	"It looks as if there is someone in the garden."											

Table 2. Map 54a in SAND1 shows four syntactic variables in the subject doubling domain.

Context	Subject doubling 2 singular											
Variables	{ V_{FINITE} __ , *__ V_{FINITE} __ , C __ , *$C_{COMPARATIVE}$ __ }											
Example	Ge	gelooft	gij	zeker	niet	dat	hij	sterker	is	as	-ge	gij.
Gloss	you_{weak}	believe	you_{strong}	certainly	not	that	he	stronger	is	than	you_{weak}	you_{strong}
Translation	"You do not seem to believe that he is stronger than you."											

Table 3. Map 68a in SAND1 shows five syntactic variables in the reflexives domain.

Context	Weak reflexive pronoun as object of inherent reflexive verb						
Variables	{ zich, hem, *zijn eigen, zichzelf, hemzelf }						
Example	Jan	herinnert	zijn	eigen	dat	verhaal	wel.
Gloss	John	remembers	his	own	that	story	[affirmative]
Translation	"John certainly remembers that story."						

Table 4. Map 84a in SAND1 shows four syntactic variables in the fronting domain.

Context	Short subject relative, complementiser following relative pronoun									
Variables	{ *1:die 2:as/at/da(t), 1:die 2:-t, 1:dien 2:at/da(t), 1:die/dat 2:wat }									
Example	Dat	is	de	man	die	dat	het	verhaal	verteld	heeft.
Gloss	that	is	the	man	who	that	the	story	told	has
Translation	"That is the man who told the story."									

6.3 Sample data illustration and diagram

Figures 1 and 2 illustrate the data mining procedure presented in the next section by defining a small subset of the actual SAND1 data. Figure 1 marks the geographical occurrences in seven Dutch dialects (1-7) of the four example variables (A-D) shown in Tables 1 to 4. For example, Figure 1 shows that in the dialects of Ouddorp (1), Merckeghem (2), Brussel (3) and Gemert (4), people can say *'t Lijkt wel of dat er iemand in de tuin staat* (A). This variable does not occur in the dialects of Nieuwmoer (5), Boskoop (6) and Nijkerk (7). Likewise, only in the village of Nieuwmoer have all of the following three variables been attested: *Als gij gezond leeft, leef-de gij langer* (B), *Jan herinnert z'n eigen dat verhaal wel* (C), and *Dat is de man die dat het verhaal verteld heeft* (D). Figure 2 shows a symbolic representation of the sample data in Figure 1. The remainder of the current article uses the symbolic variable characters (A-D) and dialect numbers (1-7) to refer to the sample data components to enhance readability.

6.4 Association rule mining based on proportional overlap

The SAND1 sample data described above are used to illustrate how relationships between variables in a database can be discovered using a technique best known as data mining but arguably more accurately described with its synonym Knowledge Discovery in Databases (KDD). Data mining is an umbrella term for various knowledge representation techniques such as association *rules*, decision *trees* and neural *networks*. Frawley et al. (1992) define data mining as the nontrivial ex-

Figure 1. This SAND1 sample marks the occurrences in seven dialects (1-7) of the syntactic variables (A-D) in Tables 1 to 4.

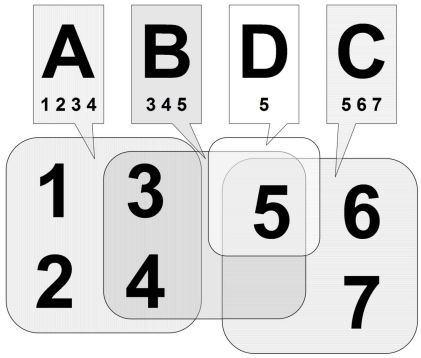


Figure 2. Symbolic representation of the SAND1 sample shown in Figure 1.

traction of implicit, previously unknown, and potentially useful information from data. Hand et al. (2001) formulate data mining more generally as the science of extracting useful information from large data sets or databases.

This work explores associations between syntactic variables in Dutch dialects using a rule induction system based on proportional overlap. Generally speaking, association rules show attribute-value conditions that occur frequently together in a given dataset. The left side of an association rule is called the antecedent and may consist of multiple predicting attributes. The right side of a rule is called the consequent and defines the predicted class(es). Association rules are typically written as 'A → C' and should be read as 'IF variable A THEN variable C'. A widely-used example of association rule mining is Market Basket Analysis, a method which examines a long list of supermarket transactions to determine which items are most frequently purchased together. It applies the Apriori algorithm to generate candidate association rules which relate the items within each transaction or basket (Agrawal et al. 1993).

$$C_n^k = C_4^3 = \binom{4}{3} = \frac{4!}{3!(4-3)!} = \frac{4\times3\times2\times1}{3\times2\times1\times(1)} = \frac{24}{6} = 4$$

Figure 3. Calculation of the number of combinations with k=3 elements from the sample data set with n=4 variables.

The application of association rule mining between syntactic variables in the current paper examines all k-combinations (or k-subsets) of syntactic variables to determine which variable subsets most frequently co-occur geographically.[3] A k-combination is an unordered collection with k unique elements.[4] Figure 3 il-

[3]The Rule INduction Console (*rinc*) programme implements the association rule mining procedure. It has been developed with the wxWidgets C++ toolkit and the next_combination STL template. The console programme is available for all software platforms and can be downloaded from http://dialectometry.net/syntax.

[4]This is in contrast with a k-permutation, which is an *ordered* collection with k unique elements.

lustrates how to calculate the binomial coefficient of the number of combinations with three elements in the sample data set of the four variables {A,B,C,D}. In this example the binomial coefficient is four and represents the combinations {A,B,C}, {A,B,D}, {A,C,D} and {B,C,D}.

Table 5. Algorithm to non-recursively evaluate all association rules.

```
1.  FOR EACH k-combination of variable set v with n elements
2.      INITIALISE combination subset s from v
3.      REPEAT
4.          FOR EACH m-combination of s
5.              INITIALISE antecedent a from s with m elements
6.              REPEAT
7.                  INITIALISE consequent c as the complement of a with k-m elements
8.                  CALL evaluateAssociationRule with a and c
9.              UNTIL all antecedent combinations a have been processed
10.         ENDFOR
11.     UNTIL all combination subsets s have been processed
12. ENDFOR
```

Table 5 lists the association rule mining algorithm in pseudocode. The procedure is scalable to even larger data sets because it is non-recursive. Therefore, memory usage remains constant. Line 1 specifies that the procedure iterates through all combinations with k=2 to k=n variables. Line 2 selects the first combination subset s with k variables. Then, lines 3 to 11 repeatedly process subset s and select the next subset. Line 4 iterates through all combinations of subset s with m=1 to m=k-1 variables. Line 5 generates the first combination subset a as the antecedent variables subset from s with m variables. Then, lines 6 to 9 repeatedly process subset a and select the next subset. Line 7 determines the corresponding consequent variables by selecting the complementary set of a from s. Finally, line 8 evaluates the quality of the generated association rule using the unique antecedent-consequent tuple based on the proportional overlap between the geographical distributions of the rule variables. The candidate association rule is accepted when it satisfies previously specified criteria of interestingness.

The procedure remains modest in automatically discarding uninteresting candidate rules. The current version of the algorithm only prunes the combination space in two cases. In the first, self-explanatory situation the interestingness value is either equal to or below zero. The second condition applies when the coverage value has the maximum value. This indicates that the antecedent encompasses the entire data set, which implies that the rule does not have any explanatory power. Of course, manual factor threshold values may be applied as well in addition to these conditions to further minimise the amount of uninteresting rules.

The proportional overlap procedure in this work consists of the following three steps. First, the lists of geographical occurrences of all syntactic variables in the rule antecedent are disjunctively merged into the rule antecedent vector of geographical occurrences. Variable occurrences are not merged conjunctively because the procedure attempts to combine microvariables to discover more general patterns. Then, the procedure constructs the rule consequent vector of geographical

occurrences. Finally, the intersection and union sets of the two vectors of geographical co-occurrences are calculated as factor components to help determine the quality of the candidate rule using a combination of indicators as listed in Table 6. The intersection set |A&C| in Table 6 represents the geographical conjunction of antecedent and consequent variable occurrences. The concept of proportional overlap is predominantly applied in research areas such as ecology and biogeography and is notably explored in (Horn 1966).

6.5 Evaluating the quality of a rule

Table 6 lists several widely used factors to help determine the quality of an association rule: accuracy, coverage, completeness and interestingness. Many more factors have been proposed over the years to further enhance rule evaluation quality. McGarry (2005) reviews a range of objective and subjective measures such as actionability, surprisingness, unexpectedness, misclassification cost, class distribution and attribute ranking, among others. These factors are not taken into account in this work. However, the current paper does incorporate complexity as the total number of variable disjuncts in both the antecedent and consequent sets. Higher complexity results are interpreted as being less interesting.

Table 6. Evaluation factors to help determine the quality of association rule A→C.

Accuracy:	|A&C| / |A|	The number of dialects which have both variables A and C divided by the number of dialects which have variable A.
Coverage:	|A| / N	The number of dialects which have variable A divided by the total number of dialects in the data set.
Completeness:	|A&C| / |C|	The number of dialects which have both variables A and C divided by the number of dialects which have variable C.
Interestingness:	|A&C| - |A||C|/N	The number of dialects which have both variables A and C minus the product of the number of dialects which have variable A with the number of dialects which have variable C divided by the total number of dialects in the data set.

It is important to note that although a pattern is expressed as a rule, it does not mean that it is true all the time. An association rule does not imply causality. The antecedent of a rule does not necessarily cause the consequent of a rule to happen. Therefore, the uncertainty in a rule should be made explicit. This is what the accuracy of a rule indicates. It signifies how often a rule is correct and is also called the confidence of a rule. The coverage of a rule expresses how often a rule applies and is also called support. The factor completeness may be used to explore how much of the target class a rule covers. This work multiplies all accuracy, coverage and completeness values by one hundred to express the rule quality factors as percentages.

The three rudimentary interestingness factors described above are always integrated in proposed measures of rule interestingness. Intuitively, rules are interesting when they have high accuracy, high coverage and deviate from the norm. The effort, then, is to formulate the optimal trade off between coverage, accuracy and potentially other factors for a specific problem domain. The domain specificity of interestingness is one of the many reasons why the ability to interactively explore

the generated association rules is always desirable and maybe even inevitable. Although data mining algorithms may use objective factors to decide whether a rule is genuinely interesting or not, domain-specific, subjective notions of interestingness may be required as well to decide whether a potentially or technically interesting rule is also genuinely interesting in a specific domain. For example, a discovered association rule may be too well-known or too trivial.

Table 7. Piatetsky-Shapiro's principles for rule interestingness (RI) measures.

1.	RI = 0 if |A&C| = |A| |C| / N.
2.	RI monotonically increases with |A&C| when other parameters are fixed.
3.	RI monotonically decreases with |A| or |C| when other parameters are fixed.

This work applies the three principles for rule interestingness measures proposed in (Piatetsky-Shapiro 1991). They are reprinted in Table 7. The principles formulate the relations between the factors accuracy, coverage and completeness as objective evaluation criteria of interestingness measures. The first principle states that the rule interestingness is zero if the antecedent and consequent of the rule are statistically independent. The second principle defines that more co-occurring elements in the antecedent and consequent of the rule will result in higher accuracy and completeness values when all other parameters remain fixed, which increases the interestingness of the rule. The third principle's interpretation is two-fold. It formulates that rule interestingness monotonically decreases with completeness when all other parameters remain fixed. Similarly, rule interestingness also monotonically decreases with coverage when all other parameters remain fixed (Freitas 1999). Note that, in contrast with accuracy, coverage and completeness values, interestingness values do not necessarily range between zero and one.

Several enhancements and alternative measures of interestingness have been proposed since (Piatetsky-Shapiro 1991). Lenca et al. (to appear) most notably describes numerous measures of interestingness in detail. The current work restricts itself to Piatetsky-Shapiro's measure of interestingness because of its historical position and formulaic simplicity. Note, however, that its symmetric nature is a property where this measure seems lacking. This is not the case for the factors accuracy, coverage and completeness. To a certain extent the influence of symmetricity can be compensated by ranking the entire result set of association rules firstly on descending interestingness, secondly on ascending complexity, thirdly on descending accuracy and finally on descending coverage.

6.6 Discovery of association rules between syntactic variables

Table 8 lists the eight most interesting association rules based on occurrences in seven dialects of the four syntactic variables in the sample data as shown in Figures 1 and 2. The algorithm in Table 5 generates fifty variable combinations for the sample data. Fourteen candidate rules are potentially interesting based on the Piatetsky-Shapiro measure of interestingness and have at least some explanatory power. From a technical perspective this means that fourteen association rules

have an interestingness value greater than 0 and a coverage value smaller than 100 percent. The list in Table 8 is sorted on descending interestingness, ascending complexity and descending accuracy, respectively.[5]

Table 8. The eight most interesting association rules in the sample data set as shown in Figures 3 and 4 sorted on descending interestingness, ascending complexity and descending accuracy.

#	*Antecedent*	→	*Consequent*	*Interestingness*	*Complexity*	*Accuracy %*	*Coverage %*	*Completeness %*
1.	B	→	A ∨ D	0.86	1	100	42	60
2.	A ∨ D	→	B	0.86	1	60	71	100
3.	D	→	B	0.57	0	100	14	33
4.	D	→	C	0.57	0	100	14	33
5.	B	→	D	0.57	0	33	42	100
6.	C	→	D	0.57	0	33	42	100
7.	B	→	A	0.29	0	66	42	50
8.	A	→	B	0.29	0	50	57	66

The list of association rules is primarily sorted on descending interestingness since the main goal of this work is to discover the most interesting association rules between the variables. The list's secondary sort factor uses ascending values of complexity which can be interpreted as an extension of the measure of interestingness. An increasing number of variable components in a rule decrease its comprehensibility and, therefore, its interestingness. Coincidentally, the application of the complexity factor in the sample data does not actually change the rule order. The list of association rules in Table 8 is ternarily sorted on descending accuracy. However, it would be equally valid to apply descending completeness as an alternative ternary sort factor. Favouring accuracy over completeness simply signifies that it is considered more important that a rule is correct than it is to discover the degree to which the consequent variables are predicted by the antecedent variables. The definitions of accuracy and completeness in Table 6 also illustrate these alternate perspectives on rule importance quite evidently. The first two rules in Table 8 demonstrate the effect of choosing completeness over accuracy to optimally sort the association rules. The rules have identical levels of interestingness and complexity but differ in the degree of accuracy and completeness. The first rule states that *if* variable B occurs in a dialect *then* variable A or D always occur as well; the rule is 100 percent accurate. However, it does not imply that the inverse is true as well. Indeed, in dialects one and two either variable A or D occurs but not variable B. This is specified in the second rule which states that if either variable A or D occurs in a dialect, then there is a 60 percent certainty that variable B occurs as well. This example adequately illustrates the asymmetric nature of the relationship between the antecedent variables and the consequent variables of an association rule. Furthermore, an asymmetric variable association may be interpreted as a variable dependency with potentially hierarchical implications.

[5]The list of potentially interesting association rules can be sorted interactively using an external software programme such as Excel or SPSS.

6.7 Data mining the Syntactic Atlas of the Dutch Dialects

The following pages highlight a small selection of potentially interesting association rules between the 485 syntactic variables in the SAND1 database based on their geographical co-occurrences in 267 Dutch dialects. The algorithm evaluated 234,740 rules without any variable disjunctions, i.e. all antecedents and consequents consist of only one variable, and found 10,730 interesting associations with an accuracy value of 90 percent or higher. This observation manifests the considerable proportional overlap between the syntactic variables in SAND1. Additionally, it could arguably be interpreted as an indication that highly interesting association rules with high coverage and high accuracy values effectively reduce the importance of the geographical occurrences in the data set. The information value of geography—by definition—becomes limited to generic density and distributional information when variable distributions overlap nearly perfectly. Ascending from the observational level of geographical distributions to more abstract variable associations would facilitate syntactic analyses to identify implicational chains and other association patterns.

Table 9. Example of a highly ranked association rule in SAND1 with one variable disjunct: "if either antecedent variable A1 or A2 occurs, then it is certain that the consequent variable also occurs".

Antecedent A1:	p46b:julle(n)/jullie (Subject pronouns 2 plural, strong forms, complex) We geloven dat julle(n)/jullie niet zo slim zijn als wij. we believe that you$_{plural,strong}$ not so smart are as we. 'We believe that you are not as smart as we are.'
Antecedent A2:	p46b:julder/jielder (Subject pronouns 2 plural, strong forms, complex) We geloven dat julder/jielder niet zo slim zijn als wij. we believe that you$_{plural,strong}$ not so smart are as we. 'We believe that you are not as smart as we are.'
Consequent:	p46a:j-[lieden-compositum] (Subject pronouns 2 plural, strong forms) We geloven dat j-lieden niet zo slim zijn als wij. we believe that you$_{plural,strong}$ not so smart are as we. 'We believe that you are not as smart as we are.'
Statistics:	Rank=9, Combination=5,327,848, Interestingness=61.31, Accuracy=100%, Coverage=40%, Completeness=93%, Complexity=1, A-Locations=107, C-Locations=114, AC-Overlap=107, AC-Disjunction=114
Interpretation:	The infrequent pronoun 'julder/jielder' perfects the implicational association of the frequent 'julle(n)/jullie' variant with the pronoun 'j-lieden'.

The number of variable combinations rises to 113,614,160 candidate rules as soon as either the antecedent or consequent of a rule may include one variable disjunction. No less than 56,267,729 generated association rules are at least 90 percent accurate.[6] This is to be expected since the algorithm disjunctively combines variables. Once a strong association between two variables has been found, any disjunctively added variable will further strengthen the association.

Table 9 presents an association rule with one variable disjunction as an example of a potentially interesting rule with a higher complexity. However, higher complexity association rules become exceedingly more difficult to interpret linguistically. As a matter of fact, it can already be quite challenging to linguis-

[6]The corresponding output file is 33 GB. The programme execution time was around 18 hours on a MacMini PowerPC G4 (1.5 GHz) computer.

tically interpret rules without variable disjunctions. Interactive explorations can only partly facilitate the evaluation process. Therefore, the remainder of the current paper concentrates on association rules without variable disjunctions.

Table 10. The most interesting rule in SAND1 without variable disjuncts.

Antecedent:	p46a:g-lieden (Subject pronouns 2 plural, strong forms) We geloven dat g-lieden niet zo slim zijn als wij. we believe that you$_{plural,strong}$ not so smart are as we. 'We believe that you are not as smart as we are.'
Consequent:	p38b:gij/gie (Subject pronouns 2 singular, strong forms) Ze gelooft dat gij/gie eerder thuis bent dan ik. she believes that you$_{singular,strong}$ earlier home are than I 'She thinks that you'll be home sooner than me.'
Statistics:	Rank=1, Combination=10,321, Interestingness=58.38, Accuracy=99%, Coverage=39%, Completeness=89%, Complexity=0, A-Locations=105, C-Locations=116, AC-Overlap=104, AC-Disjunction=117
Interpretation:	The plural pronoun 'g-lieden' belongs to the same paradigm as the singular pronoun 'gij'.

Table 10 shows the potentially most interesting association rule in SAND1 without variable disjunctions. The rule associates one of the variables in map A on page 46 in SAND1 with a variable in map B on page 38. It states that, in the context of a strong *plural* subject pronoun in second person, if the complex pronoun 'g-lieden' occurs, then the strong *singular* subject pronoun in second person 'gij' (or 'gie') nearly always occurs as well. This is indicated by the accuracy value of 99 percent. This value is calculated using the definition in Table 6 as follows: $|A\&C| / |A| * 100$ = AC-Overlap / A-Locations * 100 = 104 / 105 * 100 = 0.99 * 100 = 99 percent. Similarly, the interestingness value results as follows: $|A\&C| - |A||C|/N$ = AC-Overlap - (A-Locations * C-Locations / 267) = 104 - (105 * 116 / 267) = 104 - 45.62 = 58.38.

The geographical distributions of the rule variables in Table 10 are patterned quite coherently (not shown). All occurrences are found in the southern half of the Dutch language area. Although it may not be particularly surprising to discover a strong association between two typically southern word forms, it does not automatically follow that it may not be considered interesting or even significant to discover that the geographical overlap between, specifically, these two southern word forms is nearly all-inclusive. It is sufficient to interactively sort all association rules on antecedent name, descending interestingness and descending accuracy, respectively, to verify this hypothesis. This action reveals that only nine potentially interesting association rules exist with the complex pronoun 'g-lieden' as their antecedent and which also have an accuracy of 90 percent or higher.

The top six 'g-lieden' rules state that if in a dialect people can say *We geloven dat g-lieden niet zo slim zijn als wij* 'we believe that you$_{strong}$ not so smart are as we', then people in that dialect can also say, in descending degree of certainty, (a) *Ze gelooft dat gij/gie eerder thuis bent dan ik* 'she believes that you earlier home are than I', (b) *Ik denk da Marie hem zal moeten roepen* 'I think that Mary him will must call', (c) *U [niet-beleefdheidsvorm] gelooft dat Lisa even mooi is als Anna* 'you [non-honorific] believe that Lisa as beautiful is as Anna', (d) *Fons zag een slang naast hem* 'Fons saw a snake next to him', (e) *Erik liet mij voor hem*

werken ‘Erik let me for him work’ and (f) *De jongen wie/die z’n moeder gisteren hertrouwd is* ‘the boy who/that his mother yesterday remarried is’.

Rules (d) and (e) also strongly indicate a relationship between the second person, plural complex pronoun ‘g-lieden’ and the third person, singular, reflexive pronoun ‘hem’. It is unclear how this association should be interpreted linguistically. Although the rules might describe a previously unknown linguistic relationship, it could also merely reflect that the variables are geographically clustered. The latter case would signify the methodological reminder that a strong variable association does not necessarily imply a linguistic causation. All in all, the analysis above adequately illustrates how exploration of one association rule may easily trigger interactive investigations of several more potentially interesting rules and may raise new questions to answer.

Another approach of interactively exploring the result set of rules focuses on the examination of implicational chains between syntactic variables. Table 11 lists the highest ranked implicational chain of four syntactic variables in the set of association rules without variable disjunctions to illustrate this phenomenon. First, rule six states that if subject doubling occurs after V in second person singular, then it also appears after V in second person plural. Second, the third highest rule asserts that if subject doubling occurs after V in second person plural, then the second person plural pronoun ‘g-lieden’ nearly always arises as well. As an aside, this rule effectively demonstrates the implicit capacity to discover variable associations across syntactic domains. Third, the highest ranked rule convincingly associates the second person plural pronoun ‘g-lieden’ with the second person singular pronoun ‘gij/gie’. Finally, rule eight confirms the transitive nature of the rules with the association between subject doubling after V in second person singular and the second person singular pronoun ‘gij/gie’.

From a statistical perspective many more linguistically interesting variable associations can be expected to surface upon closer investigation. The explorations described above merely attempt to indicate the great potential of association rule mining as a meaningful contribution to linguistic theory in general and syntactic theory in particular. Another promising approach could employ association rule mining to quantitatively validate existing and new typological hypotheses. This is in contrast with the current approach which focuses on exploration and identification of variable patterns. However, every approach will require extensive consultation with syntactic theorists to meaningfully interpret the data. SAND1 provides geographical maps of many individual variable distributions to facilitate interpretation and validation of potentially interesting association rules. The generated sets of induced association rules and the rule induction programme are publicly available for interactive exploration at http://dialectometry.net/syntax/.

6.8 Conclusions

This research has successfully demonstrated how associations between syntactic variables in Dutch dialects can be discovered computationally using an association rule mining technique based on proportional overlap. The rule induction system

Table 11. The most interesting implicational chain of association rules between four syntactic variables: d54a:after_v → d55a:after_v → p46a:g-lieden → p38b:gij/gie.

Variable 1/4: d54a:after_v (Subject doubling 2 singular)

As	gij	gezond	leeft,	leef- de	gij	langer.
if	$you_{singular}$	healthily	live,	live- $you_{singular,weak}$	$you_{singular,strong}$	longer

'If you live healthily you will live longer.'
Rank=6, Combination=6,509, Interestingness=52,78, Accuracy=92

Variable 2/4: d55a:after_v (Subject doubling 2 plural)

As	gulder	gezond	leeft,	leef- de	gulder	langer.
if	you_{plural}	healthily	live,	live- $you_{plural,weak}$	$you_{plural,strong}$	longer

'If you live healthily you will live longer.'
Rank=3, Combination=7.503, Interestingness=54,07, Accuracy=93

Variable 3/4: p46a:g-lieden (Subject pronouns 2 plural, strong forms)

We	geloven	dat	g-lieden	niet	zo	slim	zijn	als	wij.
we	believe	that	$you_{plural,strong}$	not	so	smart	are	as	we.

'We believe that you are not as smart as we are.'
Rank=1, Combination=10,321, Interestingness=58,38, Accuracy=99

Variable 4/4: p38b:gij/gie (Subject pronouns 2 singular, strong forms)

Ze	gelooft	dat	gij/gie	eerder	thuis	bent	dan	ik.
she	believes	that	$you_{singular,strong}$	earlier	home	are	than	I

'She thinks that you'll be home sooner than me.'
Rank=8, Combination=6,552, Interestingness=52,73, Accuracy=98

facilitates identification and exploration of previously unknown variable relationships and validation of existing parametric generalisations. The ability to define variable associations asymmetrically is considered to be an important property of the technique in the syntactic domain. The analysis of the sample data has indicated that the Piatetsky-Shapiro measure of interestingness adequately formulates the relationships between the evaluation factors of accuracy, coverage and completeness.

The application of the association rule mining technique to the Syntactic atlas of the Dutch dialects has revealed the existence of many potentially interesting associations with high accuracy and coverage values and showed considerable overlaps between the geographical distributions of syntactic variable pairs. The exploratory review has examined the highest ranked association rules and also discussed an implicational chain of variable associations. The results strongly indicate that many more potentially interesting associations between syntactic variables are likely to be uncovered upon further investigation.

6.9 Discussion

The approach presented in this paper to discover associations between syntactic variables can be extended and refined in several ways. For example, the candidate generation algorithm listed in Table 5 could be extended to incorporate exception rules as well. These are rules which cannot be predicted from existing knowledge and typically combine high accuracy with poor coverage values. Further refinements of the data mining procedure may include experimentation with alternative measures of interestingness and incorporation of additional rule quality evaluation

factors such as surprisingness, among others.

An interesting property of data mining applications such as association rule mining arises as more variables become available to the procedure. The formula in Figure 3 shows that the number of generated candidate association rules increases factorially with the number of variables. Also, increasing complexity is another source of combinatory explosion. These observations are relevant in the current context because the second volume of the SAND (SAND2) is due to appear at the end of 2007. Incorporation of the SAND2 data into the association rule discovery process will result in a linguistic database containing around 750 syntactic variables and covering all major syntactic microvariation domains. Although the linguistically trained mind may be extremely effective in heuristically associating variables, the astronomical SAND combination space will undoubtedly exceed human limits of association precision and capacity. Additionally, the compartmented and repetitive nature of data mining algorithms makes them good candidates for computational scaling and parallellisation using grid computing techniques. Therefore, a combination of the unsurpassed human heuristic capabilities with the verifiable precision and processing power available to data mining tools may well contribute to the understanding of the structural diversity of language varieties. There is, of course, no reason to stop incorporating more data into the procedure. For example, it could be really interesting to combine available phonological data with these syntactic data to discover potential associations between variables among linguistic levels (Spruit et al. n.d.).

An entirely different application of association rule mining analyses the set of variable associations to define clusters of geographically overlapping variables known as composite variables (Spruit 2006). This application assumes that if a group of variables nearly always occur together, then a single variable of such a group does not add to the variation between two language varieties by itself. Therefore, from a quantitative perspective the cluster of variables can be interpreted as one entity which should more accurately quantify syntactic variation. Preliminary visualisations of the distance relationships between Dutch dialects based on the Jaccard distance between composite syntactic variables appear to classify the Dutch dialect areas quite accurately. The dialect maps appear to be in line with expert opinion and correspond with dialect distance visualisations (cf. (Spruit 2006, Spruit et al. n.d.)) but require further research.

Finally, it would be interesting to compare the discovered variable associations with results based on more classic statistical methods such as Cramér's V or correspondence analysis. Cramér's V is a statistic which measures the strength of association between two categorical variables based on the χ^2-statistic. Time permitting, this approach could be well worth investigating. One of the method's attractive benefits is that it calculates the statistical significance of each variable pair association. Another statistical technique which may hold promise is correspondence analysis. This method resembles the factor analysis technique but has specifically been designed to help explore associations between categorical variables. However, the interpretability of the resulting correspondence visualisations may become an issue given the considerable geographical overlaps between the

syntactic variable distributions. Furthermore, a more fundamental shortcoming of the two alternative approaches described above is the inherent symmetric nature of the discovered variable associations.

References

Agrawal, R., Imielinski, T. and Swami, A.(1993), Mining association rules between sets of items in large databases, *in* P. Buneman and S. Jajodia (eds), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, ACM Press, Washington, D.C., pp. 207–216.

Barbiers, S., De Vogelaer, G. and Devos, M.(2005), *Syntactic Atlas of the Dutch Dialects*, Vol. 1, Amsterdam University Press, Amsterdam.

Cornips, L. and Jongenburger, W.(2001), Elicitation techniques in a Dutch syntactic dialect atlas project, *in* H. Broekhuizen and T. van der Wouden (eds), *Linguistics in the Netherlands*, John Benjamins, Philadelphia/Amsterdam, pp. 53–63.

Frawley, W., Piatetsky-Shapiro, G. and Matheus, C.(1992), Knowledge discovery in databases: An overview, *AI Magazine* **13**, 213–228.

Freitas, A.(1999), On rule interestingness measures, *Knowledge-based Systems* **12**, 309–315.

Gianollo, C., Guardiano, C. and Longobardi, G.(to appear), Three fundamental issues in parametric linguistics, *in* T. Biberauer (ed.), *The Limits of Syntactic Variation*, John Benjamins, Philadelphia/Amsterdam.

Hand, D., Mannila, H. and Smyth, P.(2001), *Principles of Data Mining*, The MIT Press, Cambridge, MA.

Haspelmath, M.(to appear), Parametric versus functional explanations of syntactic universals, *in* T. Biberauer and A. Holmberg (eds), *The Limits of syntactic variation*, Benjamins, Amsterdam.

Horn, H.(1966), Measurement of overlap in comparative ecological studies, *The American Naturalist* **100**, 419–424.

Lenca, P., Meyer, P., Vaillant, B. and Lallich, S.(to appear), On selecting interestingness measures for association rules: user oriented description and multiple criteria decision aid, *European Journal of Operational Research*, Elsevier.

McGarry, K.(2005), A survey of interestingness measures for knowledge discovery, *The Knowledge Engineering Review* **20**, 39–61.

Newmeyer, F.(2005), *Possible and probable languages: a generative perspective on linguistic typology*, Oxford University Press, Oxford.

Piatetsky-Shapiro, G.(1991), Discovery, analysis and presentation of strong rules, *in* G. Piatetsky-Shapiro and W. Frawley (eds), *Knowledge Discovery in Databases*, AAAI/MIT Press, pp. 229–248.

Rizzi, L.(1986), Null objects in Italian and the theory of pro, *Linguistic Inquiry* **17**, 501–557.

Spruit, M.(2006), Measuring syntactic variation in Dutch dialects, *in* J. Nerbonne and W. J. Kretzschmar (eds), *Literary and Linguistic Computing, special issue on Progress in Dialectometry: Toward Explanation*, Vol. 21, Oxford University Press, Oxford, pp. 493–506.

Spruit, M., Heeringa, W. and Nerbonne, J.(n.d.), Associations among linguistic levels, Presented at a special session at Digital Humanities, Paris, 6 July 2006.

7

A pilot study for automatic semantic role labeling in a Dutch corpus

Gerwert Stevens[†]*, Paola Monachesi*[†]*, and Antal van den Bosch*[‡]
[†]Utrecht University
[‡]Tilburg University

Abstract

We present an approach to automatic semantic role labeling (SRL) carried out in the context of the D-coi project. Although there has been an increasing interest in automatic SRL in recent years, previous research has focused mainly on English. Adapting earlier research to the Dutch situation poses an interesting challenge especially because there is no semantically annotated Dutch corpus available that can be used as training data. Our automatic SRL approach consists of three steps: bootstrapping from an unannotated corpus with a rule-based tagger developed for this purpose, manual correction and training a machine learning system on the manually corrected data. The input data for our SRL approach consists of Dutch sentences from the D-COI corpus, syntactically annotated by the Dutch dependency parser Alpino.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

7.1 Introduction

The creation of semantically annotated corpora has lagged dramatically behind. As a result, the need for such resources has now become urgent. Several initiatives have been launched at the international level in the last years, however, they have focused almost entirely on English and not much attention has been dedicated to the creation of semantically annotated Dutch corpora.

The Flemish-Dutch STEVIN-program has identified semantic annotation as one of its priorities.[1]

Within the project *Dutch Language Corpus Initiative* (D-Coi), guidelines have been developed for the annotation of a Dutch written corpus. In particular, a 50 million word pilot corpus has been compiled, parts of which have been enriched with (verified) linguistic annotations.[2]

One of the innovative aspects of the D-Coi project is that it has focused not only on the revisions of those protocols which have been already developed within the Spoken Dutch Corpus (CGN) (Oostdijk 2002) for PoS tagging, lemmatization and syntactic annotation but it has also explored the possibility of integrating an additional annotation layer based on semantic information. This annotation layer was not present in the Spoken Dutch Corpus.

One of the goals of the D-Coi project is the development of a protocol for such an annotation layer. In particular, we have dealt with two types of semantic annotation, that is semantic role assignment and temporal and spatial semantics. The reason for this choice lies in the fact that semantic role assignment (i.e. the semantic relationships identified between items in the text such as the agents or patients of particular actions), is one of the most attested and feasible types of semantic annotation within corpora. On the other hand, temporal and spatial annotation was chosen because there is a clear need for such a layer of annotation in applications like information retrieval or question answering (Schuurman and Monachesi 2006).

Only a small part of the corpus has been annotated with semantic information, in order to yield information with respect to its feasibility. Hopefully, a more substantial annotation will be carried out in the framework of a follow-up project aiming at the construction of a 500 million word corpus, in which one million words will be annotated with semantic information.

The focus of this paper is on semantic role annotation.[3] We briefly discuss the choices we have made in selecting an appropriate annotation protocol. Furthermore, we present the results of a pilot study for automatic semantic role labeling (SRL) based on the D-coi corpus.

[1] http://taalunieversum.org/taal/technologie/stevin/
[2] http://lands.let.ru.nl/projects/d-coi/
[3] http://www.let.uu.nl/ Paola.Monachesi/personal/DCOI

7.2 Existing projects

During the last few years, corpora enriched with semantic role information have received much attention, since they offer rich data both for empirical investigations in lexical semantics and large-scale lexical acquisition for NLP and Semantic Web applications. Several initiatives are emerging at the international level to develop annotation systems of argument structure, within the D-coi project we have tried to exploit existing results as much as possible and to set the basis for a common standard. We want to profit from earlier experiences and contribute to existing work by making it more complete with our own (language specific) contribution given that most resources have been developed for English.

Within D-coi, the following projects have been evaluated in order to assess whether the approach and the methodology they have developed for the annotation of semantic roles could be adopted for our purposes:

- FrameNet (Johnson et al. 2002);
- PropBank (Kingsbury et al. 2002);

Given the results they have achieved, we have taken their insights and experiences as our starting point.

FrameNet reaches a level of granularity in the specification of the semantic roles which might be desirable for certain applications (i.e. Question Answering). However, it makes automatic annotation of semantic roles rather problematic and might raise problems with respect to uniformity of role labeling even if human annotators are involved. Furthermore, incompleteness constitutes a serious problem, i.e. several frames and relations among frames are missing mainly because FrameNet is still under development. Adopting the FrameNet lexicon for semantic annotation means contributing to its development with the addition of (language specific) and missing frames.

In our study, we have assumed that the FrameNet classification even though it is based on English could be applicable to Dutch as well. Although Dutch and English are quite similar, there are differences on both sides. For example, in the case of the Spanish FrameNet it turned out that frames may differ in their number of elements across languages (cf. Subirats and Petruck (2003) and Subirats and Sato (2004)).

Due to the limitation of available resources, the other alternative was to employ the PropBank approach which has the advantage of providing clear role labels and thus a transparent annotation for both annotators and users. Furthermore, there are promising results with respect to automatic semantic role labeling for English thus the annotation process could be at least semi-automatic. A disadvantage of this approach is that we would have to give up the classification of frames in an ontology, as is the case in FrameNet, which could be very useful for certain applications, especially those related to the Semantic Web. However, in Monachesi and Trapman (2006) suggestions are given on how the two approaches could be reconciled.

A decision was made to adopt a PropBank approach within D-coi mainly because of the prospect of semi-automatic annotation. However, the PropBank annotation guidelines needed to be revised in order to deal with Dutch constructions and with the syntactic annotation layer in D-coi.

Notice that both PropBank and D-coi share the assumption that consistent argument labels should be provided across different realizations of the same verb and that modifiers of the verb should be assigned functional tags. However, they adopt a different approach with respect to the treatment of traces since PropBank creates co-reference chains for empty categories while within D-coi empty categories are almost non existent and in those few cases in which they are attested, a coindexation has been established already at the syntactic level. Furthermore, D-coi assumes dependency structures for the syntactic representation of its sentences while PropBank employs phrase structure trees. In addition, Dutch behaves differently from English with respect to certain constructions and these differences should be spelled out. (Trapman and Monachesi 2006)

7.3 Automatic SRL

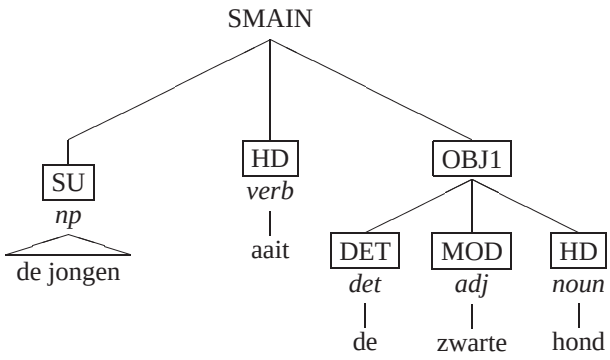
Ever since the pioneering article of Gildea and Jurafsky (2002), there has been an increasing interest in automatic SRL. However, previous research has focused mainly on English. Adapting earlier research to the Dutch situation poses an interesting challenge especially because there is no semantically annotated Dutch corpus available that can be used as training data. Furthermore, no PropBank frame files for Dutch exist.

In PropBank, frame files provide a verb specific description of all possible semantic roles and illustrate these roles by examples. The lack of example sentences makes consistent annotation difficult. Since defining a set of frame files from scratch is very time consuming, we decided to go for an alternative approach, in which we annotated Dutch verbs with the same argument structure as their English counterparts, thus use English frame files instead of creating Dutch ones. Although this causes some problems, for example, not all Dutch verbs can be translated to a 100% equivalent English counterpart, such problems proved to be relatively rare. In most cases applying the PropBank argument structure to Dutch verbs was straightforward. If translation was not possible, an ad hoc decision was made on how to label the verb.

The second problem, the unavailablity of training data, was partially solved by bootstrapping an unannotated corpus with a rule-based tagger. In short, our automatic SRL approach consists of three steps: bootstrapping from an unannotated corpus with a rule-based tagger, manual correction and finally training a machine learning system on the manually corrected data. The input data for our SRL approach consists of Dutch sentences from the D-COI corpus, syntactically annotated by the Dutch dependency parser Alpino (Bouma et al. 2000).

Another reason for adopting the PropBank approach was the abstract nature of PropBank argument labeling. Although PropBank roles are not abstract in the sense that different verbs have different role sets, roles are labeled with generic

Figure 7.1: Example CGN dependency graph



labels: $\text{ARG}_0 \ldots \text{ARG}_n$ and a fixed set of ARGMs. Such a predicate independent labeling system is an important precondition when building a rule-based system.

7.3.1 Dependency structures

Syntactic annotation of the D-Coi corpus is based on the CGN dependency graphs (Moortgat et al. 2000). A CGN dependency graph is a tree-structured directed acyclic graph in which nodes and edges are labeled with respectively c-labels (category-labels) and d-labels (dependency labels). C-labels of nodes denote phrasal categories, such as NP (noun phrase) and PP, c-labels of leafs denote POS tags. D-Labels describe the grammatical (dependency) relation between the node and its head. Examples of such relations are SU (subject), OBJ (direct object) and MOD (modifier). Figure 7.1 shows an example of a CGN dependency graph.

There are three main groups of dependency nodes: heads, complements and modifiers. Heads are phrasal heads of the encapsulating syntactic constituent, for example the head noun of a noun phrase. Complements determine the way the thematic structure of the head is interpreted. The most prominent complements are subject and direct object complements. Finally, modifiers mark such notions as time, place and quantity.

Intuitively, dependency structures are a great resource for a rule-based semantic tagger, for they directly encode the argument structure of lexical units, e.g. the relation between constituents. Our goal was to make optimal use of this information in an automatic SRL system. In order to achieve this, we first defined a basic mapping between nodes in a dependency graph and PropBank roles. This mapping forms the basis of our rule-based SRL system.

7.3.2 Mapping dependency structure nodes to PropBank labels

Mapping subject and object complements to PropBank arguments is straightforward: subjects are mapped to ARG0 (proto-typical agent), direct objects to ARG1 (proto-typical patient) and indirect objects to ARG2. An exception is made for ergatives and passives, for which the subject is labeled with ARG1.

Devising a consistent mapping for higher numbered argument is more difficult, since their labeling depends in general on the frame entry of the corresponding predicate. Since we could not use frame information, we used a heuristic method. This heuristic strategy entails that after numbering subject/object complements with the rules stated above, other complements are labeled in a left-to-right order, starting with the first available argument number. For example, if the subject is labeled with ARG0 and there are no object complements, the first available argument number is ARG1.

Examples of complements that can be numbered this way are predictive complements (*Ze schilderde het huis [rood]* 'She painted the house red') and verbal complements (*Ze lijkt [terughoudend te zijn]* 'She seems to be reserved').

Finally, a mapping for several types of modifiers was defined. Mapping modifiers consistently is a difficult task due to the fact that their meaning is often ambiguous. For example, the head word *op* ("on") in a prepositional phrase can refer to a location (*Ze loopt op straat* 'She walks on the street') or an indication of manner (*Ze loopt op hoge hakken* 'She walks on high heels'). We refrained ourselves from the disambiguation task, and concentrated on those modifiers that can be mapped consistently. These modifiers are:

- **ArgM-NEC** - Negation markers: lexical units such as *niet* (not), *nooit* (never) en *geen* (none)
- **ArgM-REC** - Reflexives and reciprocals: lexical units such as *mezelf* (myself) and *zichzelf* (oneself)
- **ArgM-PRD** - Markers of secondary predication: modifiers with the dependency label PREDM
- **ArgM-PNC** - Purpose clauses: modifiers that start with *om te* . These modifiers are marked by Alpino with the c-label OTI.
- **ArgM-LOC** - Locative modifiers: modifiers with the dependency label LD, the LD label is used by Alpino to mark modifiers that indicate a location of direction.

As was demonstrated in this section, thanks to the relational information they contain, it is possible to link PropBank labels to dependency nodes with relatively straightforward mapping rules. This property gives dependency trees an important advantage over phrase structure trees, which are commonly used in SRL systems. The next step in our approach is to implement the mapping rules in a rule-based semantic tagger.

7.3.3 XARA: a rule based SRL system

With the help of the mappings discussed above, we developed a rule-based semantic role tagger, which is able to bootstrap an unannotated corpus with semantic roles. We used this rule-based tagger to reduce the manual annotation effort. After all, starting manual annotation from scratch is time consuming and therefore expensive. A possible solution is to start from a (partially) automatically annotated corpus. This reduces the manual annotation task to a manual correction task.

The system we developed for this purpose is called XARA (XML-based Automatic Role-labeler for Alpino-trees) (Stevens 2006). XARA is able to tag a treebank in an XML format with semantic roles. In our experiments we used part of the D-Coi treebank as an input corpus. Dependency trees in this corpus are stored in the Alpino XML format. The structure of Alpino XML documents directly corresponds to the structure of the dependency tree: dependency nodes are represented by NODE elements, attributes of the node elements are the c-label, d-label, pos-tag, etc. The format is designed to support a range of linguistic queries on the dependency trees in XPath directly (Bouma and Kloosterman 2002). XPath (Clark and DeRose 1999) is a powerful query language for the XML format and it is the cornerstone of XARA's rule-based approach.

7.3.3.1 Rules

A rule in XARA consist of an XPath expression that addresses a node in the dependency tree, and a target label for that node, i.e. a rule is a *(path,label)* pair. For example, a rule that selects direct object nodes and labels them with ARG1 can be formulated as:

```
(//node[@rel='obj1'], 1)
```

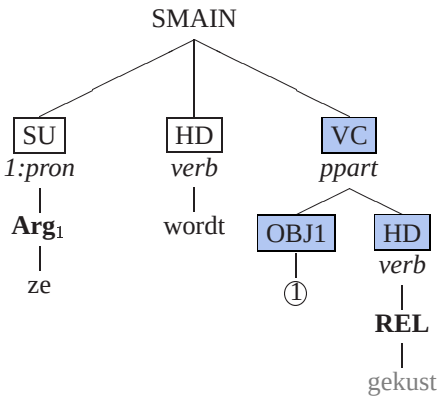
XARA supports three types of target labels. In this example, a positive integer is used. Integer labels are used to label nodes with numbered arguments (ARG_n). Secondly, for other semantic roles, such as modifiers, string values can be used. Thirdly, the special value -1 can be specified to label the target node with the first available numbered argument; this implements the heuristic labeling strategy described in the previous section.

After their definition, rules can be applied to local dependency domains, i.e. subtrees of a dependency tree. The local dependency domain to which a rule is applied, is called the rule's context. A context is defined by an XPath expression that selects a group of nodes. Contexts for which we defined rules in XARA are verbal domains, that is, local dependency structures with a verb as head. Figure 7.2 shows an example of such a context: a verbal particle. The nodes that belong to this context are dark colored.

Upon application of a rule, an attribute ("pb") is added to the target node element in the XML file. This attribute contains the PropBank label.

The combination XML + XPath proved to be a very powerful combination for the semantic annotation of our treebank. First of all, because we could work directly with the treebank files and did not need to use an intermediary format.

Figure 7.2: Example PropBank annotation on a Dependency tree



Secondly, because XPath provides a convenient and standardized method to query XML files. This enabled us to use standard Java API's. Finally, because XARA is not restricted to a specific treebank format, but can be used on any XML based treebank other than Alpino with relatively little effort. This property satisfies one of the major design criteria of the system: reusability. The only requirement is that an XML structure is used that supports XPath queries.

7.3.4 Classification system

The annotation by XARA of our treebank, was manually corrected by one human annotator. We used these manually corrected sentences as training and test data for a SRL classification system. For this learning system we employed a Memory Based Learning (MBL) approach, implemented in the Tilburg Memory based learner (TiMBL) (Daelemans et al. 2004). Memory based learning can be described as reasoning on the basis of similarity of new situations to earlier encountered situations. MBL is often categorized as a "lazy" approach to learning: instances are directly stored in memory, without any abstraction or restructuring, this is in contrast with greedy approaches such as support vector machines.

During classification, unseen examples are compared to instances in the training data. This comparison is done using a *distance metric* $\Delta(X, Y)$. The class assignment is based on the k-nearest neighbors algorithm: the most common class amongst the k most similar training instances is chosen. In case of a tie among categories, a tie breaking resolution method is used. The goal of classification is to assign class labels to a set of instances automatically. Instances represent the items to be classified by means of a set of features and their target classes.

7.3.5 Features

TiMBL assigns class labels to training instances on the basis of features. The feature set plays an important role in the performance of a classifier, and choosing features is certainly not a trivial task. In choosing the feature set for our system, we mainly looked at previous research, especially systems that participated in the CoNLL shared tasks (Carreras and Màrquez 2005) for semantic role labeling.

However, none of the systems in the CoNLL shared tasks used features extracted from dependency structures. However, Hacioglu (2004) used dependency tree features for classification. Hacioglu's system was trained and tested on data of the 2004 CoNLL shared task that was converted into dependency trees. Hacioglu classifies his approach as relation-by-relation (R-by-R) semantic role labeling. The basis of this approach is formed by a new treebank of dependency structures called DepBank. To create the DepBank corpus, first constituency trees from the Penn treebank were converted into dependency trees; furthermore, nodes in the dependency trees that cover a semantic argument were augmented with a PropBank label. For sentences with more than one predicate, the same tree was instantiated with different argument labels.

In a sense, Hacioglu's approach is comparable to our system, since in both approaches features extracted from dependency trees are used. However, there are also some differences:

- Hacioglu does not use a dependency parser to create the dependency trees, instead existing constituent trees are converted to dependency structures.
- In Hacioglu's system, a dependency tree is created for every proposition in the sentence. In our approach, labels from all propositions in a sentence are stored in a single dependency tree.
- Hacioglu only uses features that are typical to dependency trees (such as the head word of the relation). He does not use "traditional" features like phrase type, i.e. features derived from a phrase structure tree.

From features used in previous system and some experimentation with TiMBL, we derived the following feature set. The first group of features describes the predicate (verb):

(1) Predicate stem - The verb stem, provided by Alpino. This feature is analogous to the *verb lemma* feature used in many existing systems.

(2) Predicate voice - A binary feature indicating the voice of the predicate (passive/active). A predicate is considered passive if it is connected to the auxiliary verb *worden* or *zijn* and is a child of a node with c-label PPART (passive particle).

Notice that the predicate's POS tag is not used as a feature in our system, unlike in many existing systems, since all verbs in Alpino trees have the same POS tag: VERB.

The second group of features describes the candidate argument:

(3) Argument c-label - The category label (phrasal tag) of the node, e.g. NP or PP.

(4) Argument d-label - The dependency label of the node, e.g. MOD or SU.

(5) Argument POS-tag - POS tag of the node if the node is a leaf node, null otherwise.

(6) Argument position - A binary feature which indicates whether the argument is positioned before or after the predicate.

(7) Argument head-word - The head word of the relation if the node is an internal node or the lexical item (word) if it is a leaf.

(8) Head-word POS tag - The POS tag of the head word.

(9) c-label pattern of argument - The left to right chain of c-labels of the argument and its siblings.

(10) d-label pattern - The left to right chain of d-labels of the argument and its siblings.

(11) c-label & d-label of argument combined - The c-label of the argument concatenated with its d-label.

Information from this feature set that is not available to XARA is: predicate's root, label pattern of candidate argument and argument position. The position feature was added because it is was used in all CoNLL-05 systems (except one) and in the Hacioglu system. The same applies to the the predicate's root (or lemma). The label pattern feature was used in several CoNLL systems and turned out to have a positive effect on the performance of our system.

7.3.6 Training procedure

The training set consists of predicate/argument pairs encoded in training instances. Each instance contains features of a predicate and its candidate argument. Candidate arguments are nodes (constituents) in the dependency tree. This pair-wise approach is analogous to earlier work by van den Bosch et al. (2004) and Tjong Kim Sang et al. (2005).

Using every possible predicate/argument pair would result in a very large instance base that contains many irrelevant instances. This might lead to reduced performance of the classifier and low classification speed. Therefore, several methods were used to reduce the size of the instance base. The first of these methods is to ignore nodes that can never fill an argument role because of their grammatical function, for example verbal particles. The second method is to only consider phrases that are likely to be arguments.

For example, Tjong Kim Sang et al. (2005) build instances from verb/phrase pairs from which the phrase parent is an ancestor of the verb. We adopted this

approach to dependency trees: only siblings of the verb (predicate) are considered as candidate arguments.

In comparison to experiments in earlier work, we had relatively few training data available: our training set consisted of 2395 sentences. To overcome our data sparsity problem, we trained the classifier using the leave one out (LOO) method (`-t leave_one_out` option in TiMBL). With this option set, every data item in turn is selected once as a test item, and the classifier is trained on all remaining items.

Except for the LOO option, we only used the default TiMBL settings during training, to prevent overfitting because of data sparsity.

7.4 Results & Evaluation

7.4.1 Measures

We used three measures for the evaluation of our system: precision, recall and a combined measure: F-Score. Precision is defined as the proportion of predicted arguments that is predicted correctly, recall as the proportion of correctly predicted arguments. The F-Score is the harmonic mean of precision and recall. To measure the performance of the automatic systems, the automatically assigned labels were compared to the labels assigned by a human annotator.

7.4.2 Results of XARA labeling

Table 7.1 shows the performance of XARA on our treebank with 2395 sentences.

Table 7.1: Results of SRL with XARA

Label	**Precision**	**Recall**	$\mathbf{F}_{\beta=1}$
Overall	65,11%	45,83%	53,80
Arg0	98.97%	94.95%	96.92
Arg1	70.08%	64.83%	67.35
Arg2	47.41%	36.07%	40.97
Arg3	13.89%	6.85%	9.17
Arg4	1.56%	1.35%	1.45
ArgM-LOC	83.49%	13.75%	23.61
ArgM-NEG	72.79%	58.79%	65.05
ArgM-PNC	91.94%	39.31%	55.07
ArgM-PRD	63.64%	26.25%	37.17
ArgM-REC	85.19%	69.70%	76.67

Since XARA's rules cover only a subset of the argument labels, the classifier is able to achieve a much higher recall score than XARA (see table 7.2). Precision

score of the classifier is higher as well, although the difference with XARA is smaller.

Notice the contrast between XARA's performance on lower numbered arguments, especially ARG4. Manual inspection of the manual labeling reveals that ARG4 arguments often occur in propositions without ARG2 and ARG3 arguments. Since our current heuristic labeling method always chooses the first available argument number, this method will have to be modified in order achieve a better score for ARG4 arguments.

7.4.3 Results of TIMBL classification

Table 7.2 shows the performance of the TiMBL classifier on our annotated dependency treebank. This is the same treebank we used to test the XARA role labeling and consists of 2395 sentences. From these sentences, 12113 instances where extracted.

Table 7.2: Results of TiMBL classification

Label	**Precision**	**Recall**	$\mathbf{F}_{\beta=1}$
Overall	70.27%	70.59%	70.43
Arg0	90.44%	86.82%	88.59
Arg1	87.80%	84.63%	86.18
Arg2	63.34%	59.10%	61.15
Arg3	21.21%	19.18%	20.14
Arg4	54.05%	54.05%	54.05
ArgM-ADV	54.98%	51.85%	53.37
ArgM-CAU	47.24%	43.26%	45.16
ArgM-DIR	36.36%	33.33%	34.78
ArgM-DIS	74.27%	70.71%	72.45
ArgM-EXT	29.89%	28.57%	29.21
ArgM-LOC	57.95%	54.53%	56.19
ArgM-MNR	52.07%	47.57%	49.72
ArgM-NEG	68.00%	65.38%	66.67
ArgM-PNC	68.61%	64.83%	66.67
ArgM-PRD	45.45%	40.63%	42.90
ArgM-REC	86.15%	84.85%	85.50
ArgM-TMP	55.95%	53.29%	54.58

Some general observations can be made regarding these results:

- A sharp drop in precision and recall for higher numbered arguments can be observed: precision for ARG0 is 90.44%, whereas precision for ARG3 is only 21.21%. This can be contributed in part to the low number of training

examples with these labels in the corpus. Performance on lower numbered arguments is relatively good however compared to XARA's performance on these arguments.

- The ARGM label with the highest F-score is ARGM-REC. This is probably due to the fact that the only information needed to assign this label is the head word feature + POS of the head word, which makes classification of ARGM-RECs relatively easy.

- One would expect a better performance on the lower numbered arguments (assuming that the SU and OBJ1 labels are assigned accurately by the Alpino parser). We expect that the performance on these arguments can be improved by adding lexical features (see section 7.5).

It is difficult to compare our system with existing systems, since our system is the first one to be applied to Dutch texts. Moreover, our data format, data size and evaluation methods (separate test/train/develop sets versus LOO) are different from earlier research. However, to put our results somewhat in perspective, we looked at the performance of state-of-the-art SRL systems for English.

The CoNLL shared tasks provide an excellent source of information on English PropBank SRL systems that use features extracted from binary phrase structure trees. The best performing system that participated in CoNLL 2005 reached an F_1 of 80. There were seven systems with an F_1 performance in the 75-78 range, seven more with performances in the 70-75 range and five with a performance between 65 and 70.

A system that did not participate in the CoNLL task, but still provides interesting material for comparison since it is also based on dependency structures, is the Hacioglu (2004) system. This system scored 85,6% precision, 83,6% recall and 84,6 F_1 on the CoNLL data set, which is even higher than the best results published so far on the PropBank data sets (Pradhan et al. 2005): 84% precision, 75% recall and 79 F_1. These results support our claim that dependency structures can be very useful in the SRL task.

7.5 Conclusion & Further work

The results reported here, provide a first insight into the possibilities and problems of semantic role classification in a Dutch corpus based on Alpino dependency structures. Although several improvements can be made, the first results are encouraging.

One possible improvement consists in the addition of semantic features to the feature set used by the classifier. Examples of such features are the subcategorization frame of the predicate and the semantic category (e.g. WordNet synset) of the candidate argument. We expect that such semantic features will improve the performance of the classifier for certain types of verbs and arguments, especially the lower numbered arguments ARG0 and ARG1. For example, a typical type of classification error we encountered was related to verbs that can have a subject

position filled by a theme (ARG1) instead of an agent (ARG0), such as *beginnen* (”to begin”):

(1) [Het boek $_{Arg1}$] begint met een korte inleiding.
“The book begins with a short introduction”

Another example of a possible use of lexical semantic information concerns temporal and spatial modifiers (ARGM-TMP and ARGM-LOC respectively). At the moment, the only available lexical information about such modifiers in our feature set, is the head word of the corresponding preposition. In most cases however, the head word alone is not sufficient to disambiguate the preposition’s meaning. For example, the Dutch preposition *over* can either head a phrase indicating a location or a time-span. The semantic category of the neighboring noun phrase might be helpful in such cases to choose the right PropBank label. Thanks to new lexical resources, such as Cornetto (Vossen 2006), and clustering techniques based on dependency structures (Van de Cruys 2005), we might be able add lexical semantic information about noun phrases in future research.

Performance of the classifier can also be improved by automatically optimizing the feature set. The optimal set of features for a classifier can be found by employing bi-directional hill climbing (van den Bosch et al. 2004). There is a wrapper script (Paramsearch) available that can be used with TiMBL and several other learning systems that implements this approach[4]. In addition, iterative deepening (ID) can be used as a heuristic way of finding the optimal algorithm parameters for TiMBL.

Finally, it would be interesting to see how the classifier would perform on larger collections and new genres of data. The follow-up of the D-Coi project will provide new semantically annotated data to facilitate research in this area.

References

Bouma, G. and Kloosterman, G.(2002), Querying dependency treebanks in XML, *Proceedings of the Third international conference on Language Resources and Evaluation (LREC)*. Gran Canaria.

Bouma, G., van Noord, G. and Malouf, R.(2000), Alpino: wide-coverage computational analysis of Dutch.

Carreras, X. and Màrquez, L.(2005), Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling, *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2005)*. Boston, MA, USA.

Clark, J. and DeRose, S.(1999), XML Path language (XPath), *W3C Recommendation 16 November 1999*. URL: http://www.w3.org/TR/xpath.

[4]URL: http://ilk.uvt.nl/software.html#paramsearch

Daelemans, D., Zavrel, D., van der Sloot, K. and van den Bosch, A.(2004), TiMBL: Tilburg Memory Based Learner, version 5.1, reference guide, *ILK Technical Report Series 04-02*, Tilburg University.

Gildea, D. and Jurafsky, D.(2002), Automatic labeling of semantic roles, *Comput. Linguist.* **28**(3), 245–288.

Hacioglu, K.(2004), Semantic role labeling using dependency trees, *Proceedings of COLING-04*. August 2004.

Johnson, C. R., Fillmore, C. J., Petruck, M. R. L., Baker, C. F., Ellsworth, M. J., Ruppenhofer, J. and Wood, E. J.(2002), *FrameNet: Theory and Practice*.

Kingsbury, P., Palmer, M. and Marcus, M.(2002), Adding semantic annotation to the Penn Treebank, *Proceedings of the Human Language Technology Conference (HLT'02)*.

Monachesi, P. and Trapman, J.(2006), Merging Framenet and Propbank in a corpus of written Dutch, *Proceedings of the workshop Merging and layering linguistic information*. Workshop held in conjunction with LREC 2006, Genoa, Italy, 23 May 2006.

Moortgat, M., Schuurman, I. and van der Wouden, T.(2000), CGN syntactische annotatie, *Internal report Corpus Gesproken Nederlands*.

Oostdijk, N.(2002), The design of the Spoken Dutch Corpus, *in* P. Peters, P. Collins and A. Smith (eds), *New Frontiers of Corpus Research*, pp. 105–112. Amsterdam: Rodopi.

Pradhan, S., K., Krugler, V., Ward, W., Martin, J. H. and Jurafsky, D.(2005), Support vector learning for semantic argument classification, *Machine Learning Journal* **1-3**(60), 11–39.

Schuurman, I. and Monachesi, P.(2006), The contours of a semantic annotation scheme for Dutch, *Proceedings of Computational Linguistics in the Netherlands 2005*, University of Amsterdam. Amsterdam.

Stevens, G.(2006), *Automatic semantic role labeling in a Dutch corpus*, Master's thesis, Universiteit Utrecht.

Subirats, C. and Petruck, M. R. L.(2003), Surprise: Spanish Framenet!, *in* E. Hajicova, A. Kotesovcova and J. Mirovsky (eds), *Proceedings of CIL 17*. Prague: Matfyzpress.

Subirats, C. and Sato, H.(2004), Spanish Framenet and Framesql, *4th International Conference on Language Resources and Evaluation. Workshop on Building Lexical Resources from Semantically Annotated Corpora*. Lisbon (Portugal), May 2004.

Tjong Kim Sang, E., Canisius, S., van den Bosch, A. and Bogers, T.(2005), Applying spelling error correction techniques for improving semantic role labeling, *Proceedings of the Ninth Conference on Natural Language Learning (CoNLL-2005)*. Ann Arbor, MI, USA.

Trapman, J. and Monachesi, P.(2006), Manual for the annotation of semantic roles in D-coi, *Technical report*, University of Utrecht.

Van de Cruys, T.(2005), Semantic clustering in Dutch., *Proceedings of CLIN 2005*.

van den Bosch, A., Canisius, S., Daelemans, W., Hendrickx, I. and Tjong Kim Sang, E.(2004), Memory-based semantic role labeling: Optimizing fea-

tures, algorithm, and output, *in* H. Ng and E. Riloff (eds), *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*. Boston, MA, USA.

Vossen, P.(2006), Cornetto: Een lexicaal-semantische database voor taaltechnologie, *Dixit Special Issue*. Stevin.

8

Evaluating deep syntactic parsing

Using TOSCA for the analysis of why-questions

Daphne Theijssen, Suzan Verberne, Nelleke Oostdijk, and Lou Boves
Radboud University Nijmegen

Abstract

Previous research has shown that the high level of detail in syntactic trees produced by the TOSCA parsing system (Oostdijk 1996) is beneficial to *why*-question answering (QA) (Verberne et al. 2006b). TOSCA is an interactive system, i.e. it needs human verification after automatic tagging and parsing. Since only manually corrected TOSCA output has been offered to the *why*-QA system until now, TOSCA needs extrinsic evaluation of its use in the *why*-QA system. In this paper we present a necessary step towards it, namely an intrinsic evaluation of the performance of TOSCA on *why*-questions, which also enables us to trace elements in the parser that leave room for improvement. The evaluation shows that the modularity of the current TOSCA system has a dramatic effect on its performance: Tagging errors and missing syntactic markers radically decrease the coverage and the Parseval scores. Applying the Leaf-Ancestor Assessment metric for parser evaluation, we conclude that the level of detail does not really affect parser accuracy. This stimulates the automatic use of the parsing component in TOSCA for the purpose of *why*-QA. A new version of TOSCA is under construction, in which the level of detail in the parses is maintained, while there is no longer a need to separately provide POS tags or insert any syntactic markers.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

8.1 Introduction

In the field of computational linguistics parsers have been developed for generating syntactic analyses. Evaluating parser performance is useful for locating elements of the parser that leave room for improvement. If a parser is not applied for the purpose of a language technology application, evaluation is typically intrinsic, i.e. measuring the performance of a parser in the framework it is created in by comparing parser output to a truth at the right hand, a gold standard. This type of evaluation differs greatly from extrinsic evaluation, where the benefit of the parser to a language technological application is established. In this paper, we undertake an intrinsic evaluation of (the performance of) a parsing system designed for linguistic purposes – more specifically for the linguistic annotation of text corpora – that is being employed in a *why*-question answering system (Verberne et al. 2006b). In doing so, we can (1) facilitate extrinsic evaluation of the parsing system in the context of *why*-QA, and (2) formulate suggestions for a future version.

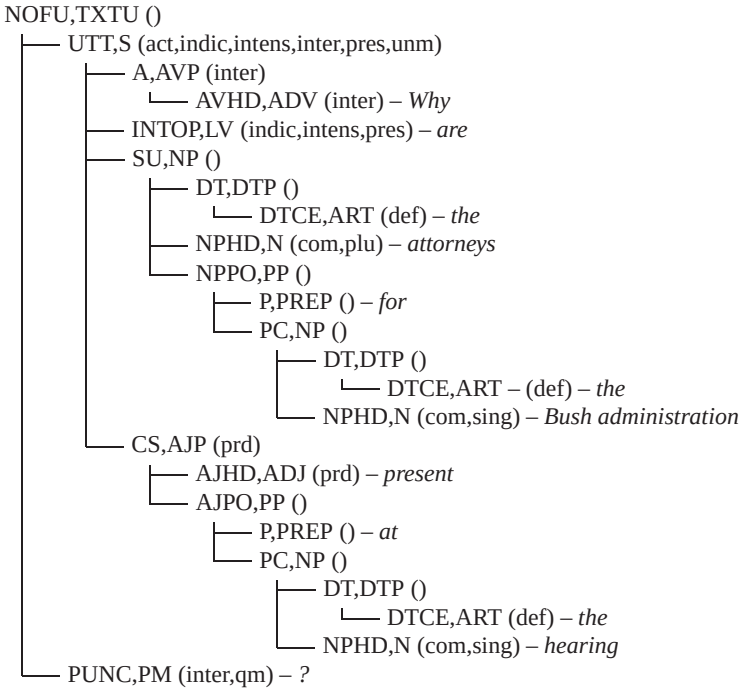


Figure 8.1: Example of TOSCA output for the question *Why are the attorneys for the Bush administration present at the hearing?*

The parser examined in the present study is the TOSCA system (Oostdijk 1996), an interactive syntactic parser that yields very detailed analyses of English text. An example of a syntactic analysis by TOSCA is presented in Figure 8.1.

TOSCA analyses are constituency trees and essentially include three types of information: (1) information pertaining to the categorial realization of constituents (e.g. article, noun phrase, clause), (2) information about the functional role of a constituent (e.g. prepositional complement, subject) and (3) additional information (for example about the word order observed or the subclass of particular word class) which is presented in the form of attributes. The three levels of analysis interfere with each other: Incorrect categories and/or attributes may lead to the erroneous assignment of function labels to constituents, for instance because the distribution of thematic roles (e.g. direct object) depends on verb transitivity.

The TOSCA system consists of two automatic components, being a part-of-speech (POS) tagger and a parser. In the current paper, the terms 'tagger' and 'parser' are only used to indicate these particular automatic components, while the total system in which both are embedded is referred to as 'TOSCA' or the '(TOSCA) system'. Input to the system are text inputs that usually take the form of sentences. These are tagged automatically with part-of-speech (POS) information. The POS tagger is probabilistic and has been trained on a manually annotated corpus. The probabilities are based on the frequency of observed word classes and the immediate context (trigrams) of each individual token in the corpus. The tag set is elaborate; it includes the basic word classes such as 'article', 'preposition', 'noun', etc., but also further subclassifications for most word classes. 'Verbs', for example, can be subdivided according to their complementation type (transitivity) and form (tense, mode and number) (van Halteren and Oostdijk 1993). The human analyst working with the system verifies whether with each of the tokens in the input string the correct tag is associated. Moreover, where required, the analyst inserts syntactic markers that help reduce the degree of ambiguity of highly ambiguous strings such as prepositional phrases and coordinated constituents. The unambiguously tagged input along with the syntactic markers that have been added is then submitted to the parser. The parser is rule-based and the formal grammar underlying it is based on the descriptive system proposed by Aarts and Aarts (1982), which is an adaptation of the English grammar by Quirk et al. (1972). Erroneously selected POS tags greatly influence the range of possible syntactic structures that can be yielded by the parser. The monotransitive form of *decline* in *Why did the Cincinnati Public schools decline to carry the program?*, for example, might be incorrectly tagged as an intransitive verb. Consequently, the clause *to carry the program* cannot be classified in any of the available syntactic structures because the verb attribute 'intransitive' prevents the assignment of the correct function to this direct object. Since the parser has no knowledge of the contextual (i.e semantic, pragmatic and extra-linguistic) knowledge that is called upon, it generates all possible syntactic analyses. However, it includes a penalty system that favours certain intuitively more appropriate analyses than others. It prefers, for example, unmarked word order over marked word order. Still a number of parses with equal penalties may remain, from which the human analyst is expected to select the one correct analysis for storage in a linguistic database. For more details on the TOSCA system, the reader is referred to van Halteren and Oostdijk (1993).

Previous research has already indicated that the deep linguistic information

provided by the TOSCA parses is useful for the *why*-question answering system currently under construction at the Radboud University Nijmegen (Verberne et al. 2006b). Until now only manually corrected TOSCA output has been offered to the system. The intrinsic evaluation presented here is the first step towards a necessary extrinsic evaluation of the use of TOSCA in the *why*-QA system. Consequently, the data used consists of *why*-questions solely. For the purpose of discovering items open to improvement, the pipelined design of the TOSCA system and the descriptive model of the parser need to be evaluated. Therefore, the aim of the present study is two-fold:

1. to evaluate the separate stages in the analysis process (POS tagging, tag selection and marker insertion, parsing and parse selection) and the way in which errors in one stage affect subsequent stages;

2. to evaluate the descriptive model used by the TOSCA grammar (incl. categories, functions and attributes and the interaction between these).

The structure of this article is as follows: The evaluation of the separate stages used in arriving at the contextually appropriate analysis for a given string is presented in section 8.2. Section 8.3 concerns the evaluation of the descriptive model of the grammar underlying the parser. Section 8.4 contains our overall conclusion and suggestions for future research.

8.2 Evaluation of separate parser modules

In the introduction the TOSCA system has been described as an interactive system. In this section we investigate the performance obtained in the different stages in the analysis process: (1) automatic tagging, (2) manual tag correction and syntactic marker insertion, (3) automatic parsing and (4) manual parse selection. To establish the effect of inaccuracies on subsequent steps, we skip over the stages requiring human intervention and evaluate the eventual parser output.

8.2.1 Data

As mentioned in the introduction, the data set consists of *why*-questions solely. *Why*-questions can be defined as interrogative sentences with the interrogative adverb *why* or one of its synonyms in (near) initial position. Despite the fact that several data sets have been developed for the purpose of question answering (QA), none of them was suitable for developing and testing a system for *why*-QA. Therefore, Verberne et al. (2006a) developed a data set by asking native speakers of English to formulate *why*-questions to thirteen different newspaper texts, with the explicit mention that the answer to the question should be present in the text. We decided to use a subset of these data for the evaluation of TOSCA. Of the first six texts we included all 138 unique questions in our data set, supplemented with another 100 questions randomly selected from the other seven texts, thus leading to a data set consisting of 238 questions. It was not feasible to use all available *why*-questions because creating gold standard parse trees is very time consuming.

For the purpose of evaluating the separate contributions of the system's components, we derived three data sets from the 238 questions: (1) a gold standard (from now on referred to as 'GOLD'), (2) a semi-automatic output ('SEMI'), in which we applied tag correction and manual insertion of syntactic markers, and (3) a fully automatic output ('AUTO'), in which only the two automatic components (POS tagger and parser) are used. Using the interactive TOSCA system we developed GOLD. For questions that could not be parsed despite our intervention after the tagging and parsing stages, we manually created gold standard trees. SEMI has been obtained by employing the interactive TOSCA system as it was meant up until the actual parsing process. Often, the parser proposed more than one possible syntactic analysis. The order in which these parses are presented is not based on linguistic theory but depends on the system's procedure of passing through the grammatical rules. For SEMI, we always saved the first proposed tree, which is neither ranked first nor completely randomly selected by the parser. To create AUTO, the list of tags proposed by the POS tagger and the first tree proposed by the parser were left unchanged. In this set-up no syntactic markers are inserted because this would involve changing the system (the insertion of syntactic markers presently requires manual intervention on the part of the human analyst; the alternative of producing a script that guesses the location of the markers would be possible, but would alter the system).

8.2.2 Method

SEMI and AUTO can be used for evaluation of the separate stages in the analysis process. An investigation of the outputs obtained while having the system operate fully automatically enables us to establish the effect that omitting tag correction and marker insertion, and refraining from parse selection (ranking) has on the eventual parser output, or put differently, the implications of corruptions in the parser input on parser performance.

We evaluate both the coverage of the parser and the quality of its output. In order to measure the robustness of the parser, we calculate the proportion of questions for which the parser was able to produce output (the coverage) for both SEMI and AUTO. We then try to find explanations for uncovered questions. In order to measure the quality of the parser output, we use Parseval, which is the common metric for evaluation of the quality of constituency trees. Parseval is also referred to as GEIG (Black et al. 1991). Parseval's evaluation method is based on lining up the brackets delimiting constituents. A sentence *a b c* with a gold standard *[a b] c* for instance, is considered not structurally consistent with an output *a [b c]*, because there is a crossing error (Black 1993). In addition to the average number of crossing brackets, precision and recall are calculated. The precision is a ratio of the number of correct brackets in the system's parse to the total number of brackets in the system's parse, while the recall is a ratio of the number of correct brackets in the system's parse to the total number of brackets in the gold standard. Following van Rijsbergen (1979), the F-score can be calculated, which represents the harmonic mean of precision and recall. Since Black (1993), the Parseval metric has

been extended. Magerman (1995) has decided to include the assignment of labels in the metric. For example, if the gold standard is *[PP [P a] [NP b]] [VP c]* and the parser output *[ADV a] [VP [V b] [V c]]*, the evaluation is based on comparisons of the location of the brackets as well as the choice of labels. This has led to the measures 'labelled precision' and 'labelled recall'.

Drawbacks of the Parseval metric are that it tends to favour minimal structure (Carroll et al. 1998) and that misattachments are penalised more than once (Lin 1995). The former can be explained by the fact that the more brackets there are, the more errors can be made. For the latter the reader is referred to Lin's (1995) example on PP-attachment, where a single error is penalised three times. The objections to Parseval have led to the development of various depencency-based parse evaluation methods (e.g. Lin 1995, Carroll et al. 1998). Since TOSCA is a constituency-based parser, the TOSCA output would have to be transformed into a uniform format convenient for the method used if we decided to use a dependency-based evaluation method. This would increase the risk of making errors and thereby decrease the performance reached. Furthermore, most of the rich syntactic information provided by TOSCA will be lost in the transformation, while we intend to use an evaluation method capable of dealing with the high degree of detail in the trees. Therefore, dependency-based methods are not suitable for the present evaluation. Fortunately, the Parseval metric has benefits that justify its use for the present evaluation, namely the fact that it is commonly employed in parser evaluation and that it enables dealing with the three types of information provided by TOSCA, as previously mentioned: categories, functions, and attributes. Following Parseval, we are able to determine the average number of crossing brackets and the labelled precision, recall and F-score for all three types separately, and average them. Averaging seems the best method to get to a single score, because multiplying the scores would penalise related errors more than once. By comparing the scores obtained for SEMI and AUTO, we can draw conclusions on the influence of errors in separate components on the eventual system output.

8.2.3 Results

The coverage, the number of perfect matches and the Parseval scores are presented in Table 8.1. From the set of 238 questions, TOSCA was able to parse 233 in SEMI, and only 190 in AUTO. Of 233 questions in SEMI, 188 were a perfect match to GOLD, compared to only 41 of 190 trees in AUTO. AUTO achieves a lower precision and recall and has more crossing brackets than SEMI (the differences in Parseval scores are significant (p=0.000) following the independent t-test). In AUTO, 84.5% of the POS tags including their specifications (e.g. *V(intr, inf)*) is completely correct for this data set.

8.2.4 Discussion

The differences between SEMI and AUTO in Table 8.1 demonstrate that the accuracy of the tags provided to the parser is essential for the performance of the

Table 8.1: Tag accuracy, coverage, perfect match and Parseval scores for SEMI and AUTO

	SEMI	AUTO
Tag accuracy	1.000	0.845
Coverage	0.979 (233 of 238)	0.798 (190 of 238)
Perfect match	0.807 (188 of 233)	0.216 (41 of 190)
Labelled Precision	0.960	0.794
Labelled Recall	0.957	0.772
Labelled F-value	0.959	0.783
Average nr crossing brackets	0.060	0.310

TOSCA system. This is obvious since the parser is designed so as to produce (minimally) the correct parse on the basis of correctly tagged input. Erroneously tagged input will cause the parser to fail to produce a correct parse. Thus human intervention is required to manually correct any erroneous tags resulting from the application of the POS tagger.

In more than 80% of the covered questions in SEMI, there is no need for the human analyser to select the correct syntactic tree, since it is presented first (0.807 perfect match). Taking into account the fact that the parser does not include a ranking procedure for trees that have obtained equal penalties during the parsing process, we consider this percentage of perfect matches rather large. It encourages a fully automatic use of the parser (i.e. the second automatic component of the TOSCA system) for the purpose of *why*-question answering.

Despite the fact that the parser is a wide-coverage parser intended to parse unrestricted input, we found that for 5 questions in our data set it was unable to produce an analysis, even when provided with gold standard tags (SEMI). Two questions included a coordination that apparently was too complex, another two were problematic because of the percent symbol (*%*) and one question included a date (*April 26, 1990*). For AUTO, the same problems occurred, except for the last-mentioned, where a tree could be produced due to tagging errors. However, in AUTO another 44 questions could not be parsed:

1. In 24 questions (54.5% of 44 uncovered questions), the proposed POS-tags caused problems with the verb phrase. In some cases, the lexical verb was not tagged as such and therefore was regarded missing by the parser. For example, in the question *Why did hundreds of thousands of people* ***march*** *in Washington twice this year?*), the lexical verb *march* was erroneously tagged as a noun. This leads to serious complications in function assignment. Other problems in the verb phrase concerned the lack or surplus of finite verbs and the inconsistency between the auxiliary and the tense of the lexical verb.

2. The lack of syntactic markers caused problems with coordination in 9 questions (20.5%), for example *Why is the decision expected by late June* ***or*** *early July?*, where the coordinated elements *late June* and *early July* were not recognised as such by the parser because of the missing marks.

3. In 8 questions (18.2%) there were problems with arguments and complements that were not caused by the incorrect tagging of verbs. These cases included instances where nouns were tagged as adverbs or adjectives causing problems in subjects and in prepositional phrases. Moreover, in some cases, a word was incorrectly tagged as a subordinating conjunction, expecting a clause while there was none, as in *why don't they like **that** idea?*

4. In 3 questions there were problems with existential *there* (it was tagged as a general adverb which was not possible at that location given the context), for instance in *why is **there** resistance to the Classroom Channel?*

The Parseval scores in Table 8.1 are significantly lower for AUTO than for SEMI, meaning that the TOSCA analyses in this case are more erroneous. Taking into consideration this finding and also the coverage, we can conclude that the parser can only perform well if it is provided with accurate input. The parser is not very robust in handling tagging errors and missing markers. As we observed above, the parser will definitely fail to produce the correct analysis if provided with incorrect or incomplete input, while in some cases there will be no output at all. Thus inaccurate input is always fatal when it comes to parse selection/ranking.

For this particular data set consisting of *why*-questions only, the accuracy of the input could be improved by training the (probabilistic) tagger on a large corpus of *why*-questions, and guessing syntactic markers by use of a script. The benefit of such solutions, however, depends on the size and uniformity of the data set concerned. It is worthwhile to establish whether a different design of the parser performs better, for example an integrated system in which the parser operates on raw input and has direct access to a lexicon, rather than a highly modularised system where POS tagging and tag selection are separate steps which are executed independently of the parser. In such a design there would be no need for human intervention since the parser would be able to negotiate the correct word class tags for the tokens in the input all by itself. Presently, such a system is being developed. The new TOSCA system is still designed to produce syntactic annotations for unrestricted (correct) English which should include the one contextually appropriate analysis for a given input string. Since more than one analysis may be produced by the parser, the system also includes a selection tool which the human analyst can use to make the appropriate selection.

8.3 Evaluating the descriptive model: categories, functions and attributes

As mentioned earlier, the TOSCA parser produces detailed syntactic analyses, indicating categories, functions and attributes. In this section we investigate the parser accuracy on all three types of labels, taking into consideration that the types are interrelated. For example, if the transitivity associated with the verb is incorrect, the subsequent assignment of syntactic roles is bound to be problematic (the parser will either fail completely or at least fail to assign the correct function labels). Investigating how accurate the parser is with each of the types of information helps us in establishing whether the level of detail of the parser output does

not lead to more complications than benefits. In this way, we are able to evaluate the descriptive model of the grammar underlying the TOSCA parser.

8.3.1 Data and method

For the evaluation of the different levels of information produced by the TOSCA parser we use the SEMI data we created for the evaluation of the pipelined design of the whole TOSCA system in the previous section. This data set consists of the 233 questions for which the parser was able to produce output. Moreover, we reuse the gold standard (GOLD) we have already developed.

The Parseval metric applied in section 8.2 provides us with several quality scores for each question, but is not helpful in pinpointing where exactly the errors are made. Therefore, we employ the approach proposed by Sampson et al. (1989), and further discussed by Sampson (2000), which is Leaf-Ancestor Assessment (LA). A possible drawback of applying different metrics of evaluation is that their notions of the degree of correctness can vary from question to question, i.e. a question can reach a high score in the one metric and a rather low in the other. Sampson and Babarczy (2003) have compared the Parseval labelled F-score and the LA score and concluded that there is only a small correlation. However, we will show in the next section that the judgements of the two metrics are highly correlated for our data set of 233 *why*-questions. An explanation for the fact that the two metrics are more similar for our data set than they were for Sampson and Babarczy's (2003) data is that our data set is more uniform because all instances are *why*-questions. The high correlation allows us to employ LA here without running the risk of presenting results that largely diverge from those presented in the previous section.

The calculation of the LA score can best be explained by means of an example. Figure 8.2 shows a syntactic tree of the question *Why are 4300 additional teachers required?*, in which *4300 additional* and *teachers* have been incorrectly analysed as two separate NP's.
Starting from a terminal element, i.e. a leaf in the tree, one moves up in the tree and registers each node label of the desired information level until one reaches the root of the tree. If necessary, squared brackets are inserted in the label sequence to delimite branches with multiple nodes. For *4300 additional*, for example, the category label sequence is *NUM NP S TXTU*. Similarly, a category label sequence can be determined for *4300 additional* in the correct syntactic analysis, which should include brackets because *4300 additional teachers* is a multi-node branch: *NUM [NP S TXTU*. The two label sequences are then compared by applying the minimum edit distance, where deletion and insertion have a penalty of 1, and substitution a penalty of 2. The minimum edit distance for the two label sequences mentioned is 1 (being a deletion of the bracket). The LA score is calculated by subtracting the minimum edit distance from the total number of labels (including brackets) in output and gold standard together, and dividing this again by the total number of labels and brackets. In the example the LA score is (9-1)/9 = 0.89. Combining the scores for all terminal elements indicates the score for the whole sentence.

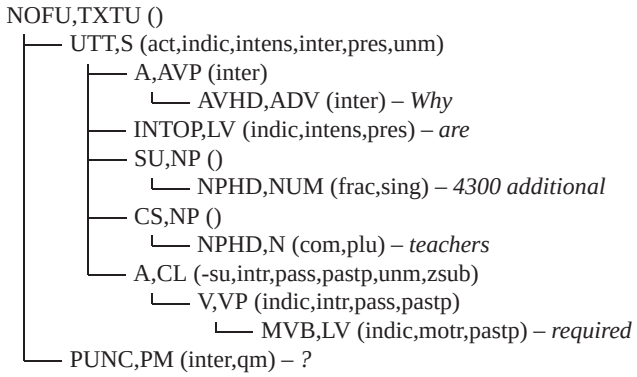


Figure 8.2: Example of a syntactic tree: *Why are 4300 additional teachers required?*

Likewise a score can be determined for the whole data set.

A disadvantage of this metric is the fact that errors in nodes high in the tree, dominating many words, have more influence on the scores than errors in lower nodes, dominating fewer words (Sampson et al. 1989, Sampson 2000). The benefit of Leaf-Ancestor Assessment is two-fold. Firstly, we use the minimal edit distance component in the LA metric for (1) analysing the tree structure, and (2) analysing the selection of categories, functions and attributes. Insertions and deletions indicate that there are too few or too many nodes in the tree, denoting incorrect tree structure. For example, there are too many labels for the verb *required* in Figure 8.2 due to the fact that it has incorrectly been parsed as a separate clause. Substitutions involve instances where nodes have been labelled incorrectly. For example, if the attribute 'passive' occurs instead of 'active', this is a label error in passivity within the attribute type of information. Secondly, the LA scores obtained for individual words or compounds can be used for listing those that fail most often, i.e. those that have the highest proportion of scores lower than 1 (1 being a perfect score). This helps in locating errors as well.

8.3.2 Results

Figure 8.3 shows a comparison between the Parseval labelled F-score and the LA score for our data set, following the example in Sampson and Babarczy (2003). The figure shows the scores for categories in the TOSCA output based on manually verified tagged input (SEMI). The focus is on categories since those are the labels that most other syntactic parsers produce. The correlation between the scores is very high (0.94). Thus, contrary to conclusions in Sampson and Babarczy (2003), both scoring metrics are highly correlated for our data set. The similarity provides us with enough support to use either method, depending on which suits the evaluation purpose best.

The LA scores for the TOSCA parses in SEMI are presented in Table 8.2. The

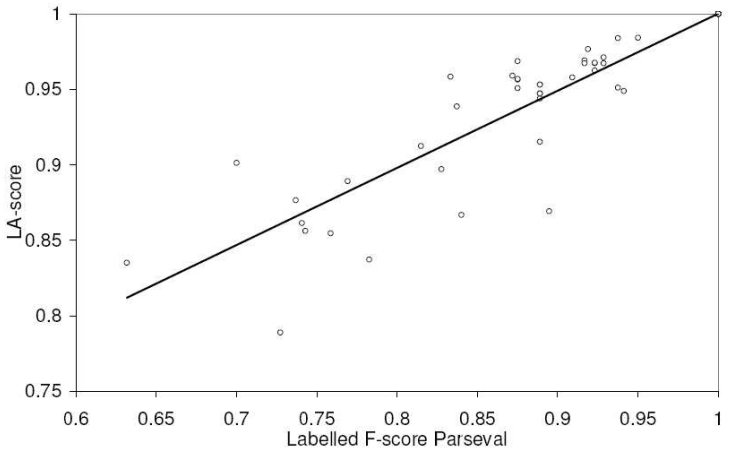


Figure 8.3: Scores for 233 TOSCA parses based on edited input (SEMI), calculated by the two metrics: Labelled F-score following Parseval and LA-score following Leaf-Ancestor Assessment

differences between the scores for categories, functions and attributes are significant (p = 0.000 for all three pairs, following the paired (dependent) t-test). The scores for categories are highest, those for functions lowest. As established in the previous section, more than 80% (188 questions) of the parses are a complete match of the gold standard.

Table 8.2: LA scores for TOSCA output in SEMI

	Categories	Attributes	Functions	Average
LA Score	0.988	0.983	0.976	0.982

Table 8.3 shows a list of words obtaining an LA score lower than 1 (being the perfect score). The first number shows the proportion of occurrences with an erroneous label sequence and the second the average LA scores obtained for all occurrences of the word. The LA scores are the average of the scores obtained for category, function, and attribute(s). We have only listed words that have a frequency of at least five, of which at least a quarter has an imperfect label sequence. This decision prevents inclusion of unique or rare words that have an imperfect analysis: if a word occurs only once in the data set and its label sequence contains an error, 100% of this word fails, which would undesirably position it high in the list.

Table 8.3: Words with an imperfect label sequence.

word	prop.	LA	word	prop.	LA	word	prop.	LA
than	0.60	0.80	*dictionary*	0.40	0.89	*at*	0.30	0.88
chefs	0.60	0.82	*with*	0.38	0.76	*women*	0.29	0.70
for	0.47	0.77	*about*	0.33	0.83	*and*	0.29	0.88
court	0.44	0.80	*warming*	0.33	0.88	*up*	0.25	0.81
supreme	0.43	0.79	*rights*	0.33	0.88	*in*	0.25	0.84
easier	0.40	0.68	*global*	0.33	0.91			

8.3.3 Discussion

There are two indicators of tree structure in the LA metric, being the position of brackets and the number of labels in the label sequences for each terminal element. In 36 questions of the 233 in the data set, there was an error in the placing of brackets. Brackets are only placed when a node has one or more sisters, so an incorrect placement of brackets is a straightforward clue for erroneous tree structure. The other sign of imperfect tree structure is the lack or surplus of node labels in a sequence. This was the case in the same 36 questions plus one other.

An example of an incorrect analysis is that yielded for the question *Why are films planned for release only overseas?*, in which *planned ... overseas* is incorrectly parsed as a postmodifier of the noun *films* (figure 8.4). The word *planned*, for instance, shows that the use of brackets fails and there is a lack of nodes (for categories: *LV VP [CL NP S TXTU* versus the gold standard *LV VP S TXTU*). Both observations help in establishing that the tree structure is erroneous and in locating in what part of the tree the errors occur.

Substitutions of node labels demonstrate incorrect label selection. They especially occur for the attributes and functions selected by the TOSCA parser and to a less extent for categories. In 7 questions, the clause tense was incorrect, for instance by mistaking a progressive construction for a present participle construction. In a few other questions (3), there were problems concerning modality or voice. Errors in the functions ‘subject’, ‘subject complement’, ‘direct object’ and ‘adverbial’ occur in 28 questions. Of these 28 questions, the transitivity of the main clause (*UTT,S* was wrong in 5 questions, in all of which a monotransitive main clause was erroneously parsed as an intransitive one. Since the parser was offered manually checked tags, the transitivity of the verb in the parse must be correct. The problem is that the monotransitive verb is erroneously placed in a subclause, making the subclause monotransitive and the main clause intransitive. This again leads to an erroneous assignment of the function labels ‘subject’ and ‘adverbial’ to elements in the non-existent subclause. In 9 questions, the question word *why* was incorrectly parsed as a subject complement instead of an adverbial. Because of this the word order feature ‘pre-cs’ instead of ‘unmarked’ is selected, meaning the fronting of a subject complement. The remaining 14 questions involving the functions mentioned have too diverse causes to describe them here.

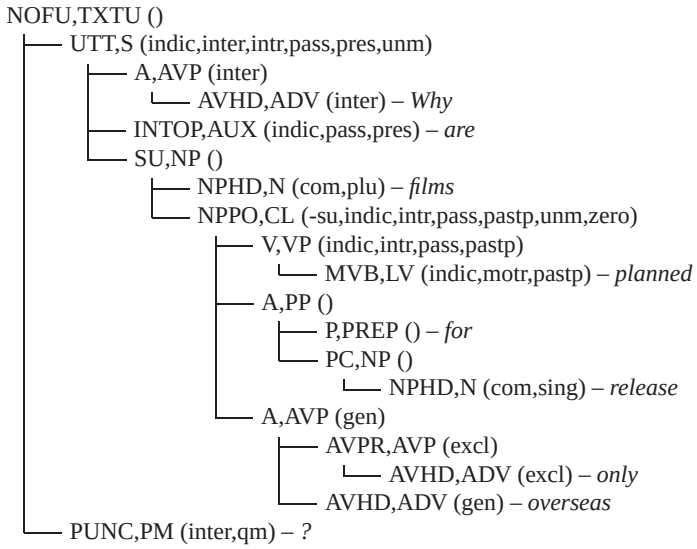


Figure 8.4: Example of TOSCA output in SEMI for the question *Why are the attorneys for the Bush administration present at the hearing?*

The word list in Table 8.3 enables us to locate difficulties in parsing the data we used. Interesting is the large number of prepositions in this list despite the fact that for the greater part, PP-attachment is determined by syntactic markers that we manually inserted prior to the parsing process. The list also shows word groups that occur in the same questions. The words *dictionary*, *easier* and *than*, for instance, are all used in questions posed to a newspaper text about compiling a Spanish equivalent of the Oxford English Dictionary (OED). It appears that though formulated by different native speakers of English, the questions have a similar structure. This is likely to be caused by the design of the elicitation experiment, where participants had access to the news paper texts while formulating questions to them. In questions to other texts, co-occurring words are *court*, *supreme*, *rights* and *women*, and *warming* and *global*. Employing a larger data set with more syntactic and lexical diversity to verify whether the results at the word level are representative for *why*-questions in general is beyond the scope of the present evaluation.

Due to the level of detail in the TOSCA output, it is difficult to compare the results to those obtained by other parsers and to establish a baseline. Often parsers only provide categories in their hierarchical structures, which is also the information level on which TOSCA reaches the highest LA scores. Functions are not commonly included in syntactic analyses due to the fact that they are less obvious to determine. This is confirmed by the lower LA scores for functions that have been obtained by TOSCA. Although a comparison with either other parsers or a

baseline cannot be made and not all three levels of information are equally successful, we assume that the LA (0.982) and perfect match scores (0.807) are sufficient to continue the use of the present descriptive model in future versions of the TOSCA parser. Furthermore, previous research has shown that the level of detail of the TOSCA trees is beneficial to the *why*-question answering system (Verberne et al. 2006b), and the presented results encourage the use of the automatic parser in the *why*-QA system.

8.4 Conclusion and further research

In this paper we have presented an intrinsic evaluation of the TOSCA system, which enabled us to pinpoint difficulties in the system and to formulate suggestions for a future version of TOSCA. Moreover, the use of *why*-questions as data facilitate the extrinsic evaluation of TOSCA in the *why*-question answering system.

TOSCA is an interactive parsing system that aims to yield deep linguistic analyses. The output includes detailed syntactic information in the form of categories, functions and attributes. The level of detail and the interdependence between the different types of information in the descriptive model that is being used entails the risk of causing a domino effect in which incorrect categories and/or attributes lead to the erroneous assignment of function labels to constituents. When provided with correct POS tags and post-edited input, however, more than 80% of the first proposed TOSCA analysis is a perfect match of the gold standard. The parses obtain an average LA score of 0.982. We consider the evaluation results sufficient to assume that the level of detail does not really affect the parse accuracy, and is therefore justified in a future version of TOSCA as well.

The modularity of the current TOSCA system is fatal: Tagging errors and missing syntactic markers in automatically obtained input radically decrease the coverage, showing that the parser is not at all robust. Moreover, the Parseval labelled F-scores for those questions that could be parsed were much lower (0.783) than those reached when the tags are corrected and the necessary markers are inserted (0.959). A new version of TOSCA is under construction, in which the level of detail in the parses is maintained, while there is no longer a need to separately provide POS tags for the tokens in the input or insert any syntactic markers.

Since the principle adopted in parsing - yielding minimally the one correct analyses for a given input string - is held onto also with the new implementation of the TOSCA system, the ranking of syntactic parses remains a topic of interest. Future research should be directed at investigating whether and how it would be possible to rank the parses in such a way that the contextually appropriate one is presented as the first one. A possible method to consider is the use of the outcome of the parser evaluation applying the Parseval or LA metric. Each presented parse could then be compared to the gold standard and ranked according to its accuracy score. Subsequently, machine learning algorithms could be employed to find patterns on which general rules for parse ranking can be based. However, such an approach demands a large annotated corpus that is not available at present and

should therefore be constructed for this purpose.

References

Aarts, F. and Aarts, J.(1982), *English Syntactic Structures*, Pergamon (Oxford).

Black, E.(1993), Statistically-based computer analysis of English, *in* E. Black, R. Garside and G. Leech (eds), *Statistically-driven computer grammars of English: The IBM / Lancaster approach*, pp. 1–16.

Black, E., Abney, S., Flickenger, S., Gdaniec, C., Grishman, C., Harrison, P., Hindle, D., Ingria, R., F., J., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B. and Strzalkowski, T.(1991), Procedure for quantitatively comparing the syntactic coverage of English grammars, *Proceedings of the workshop on Speech and Natural Language*, Leiden, pp. 306–311.

Carroll, J., Briscoe, E. and Sanfilippo, A.(1998), Parser evaluation: a survey and a new proposal, *Proceedings of the International Conference on Language Resources and Evaluation*, Granada, pp. 447–454.

Lin, D.(1995), A dependency-basedmethod for evaluating broadcoverage parsers, *Proceedings of the IJCAI-95*, Montreal, pp. 447–454.

Magerman, D.(1995), Statistical decision-tree models for parsing, *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Morgan Kaufmann, Cambridge, pp. 276–283.

Oostdijk, N.(1996), Using the TOSCA analysis system to analyse a software manual corpus, *in* R. Sutcliffe, H. Koch and A. McElligott (eds), *Industrial Parsing of Software Manuals*, Rodopi Amsterdam, pp. 179–206.

Quirk, R., Greenbaum, S., Leech, G. and Svartvik, J.(1972), *A Grammar of Contemporary English*, Longman (London).

Sampson, G.(2000), A proposal for improving the measurement of parse accuracy, *International Journal of Corpus Linguistics* pp. 53–68.

Sampson, G. and Babarczy, A.(2003), A test of the leaf-ancestor metric for parse accuracy, *Journal of Natural Language Engineering* pp. 365–380.

Sampson, G., Haigh, R. and Atwell, E.(1989), Natural language analysis by stochastic optimization: a progress report on project APRIL, *Journal of Experimental and Theoretical Artificial Intelligence* pp. 271–287.

van Halteren, H. and Oostdijk, N.(1993), Towards a syntactic database: The TOSCA analysis system, *in* J. Aarts, P. de Haan and N. Oostdijk (eds), *English Language Corpora: design, analysis and exploitation*, Rodopi (Amsterdam), pp. 145–161.

van Rijsbergen, C.(1979), *Information Retrieval*, 2nd edition, Butterworths (London).

Verberne, S., Boves, L., Oostdijk, N. and Coppen, P.(2006a), Data for question answering: the case of why, *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Verberne, S., Boves, L., Oostdijk, N. and Coppen, P.(2006b), Exploring the use of linguistic analysis for why-question answering, *Proceedings of the 16th meeting of Computational Linguistics in the Netherlands (CLIN 2005)*, Amsterdam, pp. 33–48.

9

The automatic generation of narratives

Mariët Theune[†]*, Nanda Slabbers*[†]*, and Feikje Hielkema*[‡ 1]
[†]University of Twente
[‡]University of Aberdeen

Abstract

We present the Narrator, a Natural Language Generation component used in a digital storytelling system. The system takes as input a formal representation of a story plot, in the form of a causal network relating the actions of the characters to their motives and their consequences. Based on this input, the Narrator generates a narrative in Dutch, by carrying out tasks such as constructing a Document Plan, performing aggregation and ellipsis and the generation of appropriate referring expressions. We describe how these tasks are performed and illustrate the process with examples, showing how this results in the generation of coherent and well-formed narrative texts.

9.1 Introduction

Most natural language generation (NLG) systems are aimed at 'serious' applications such as the generation of weather reports, instructions, descriptions of museum artifacts, etc. The automatic generation of narratives, however, is still a largely unexplored subject. A notable exception to this is the work by Charles

[1]Feikje Hielkema carried out this work while she was at the University of Groningen.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

Callaway on STORYBOOK (Callaway 2000), a full-fledged NLG system for narrative prose generation that can generate many different retellings of the same story (Little Red Riding Hood). The input for STORYBOOK consists of a number of plot arcs selected using user-specified parameters; the system was never coupled to a digital storytelling system that could generate original plots. Other work addressing the generation of narratives is that by Lönneker (2005), who proposed an architecture for a "narratologically enhanced NLG system" to be used in combination with a story (plot) generator. However, this architecture has not been implemented. One of few systems that have been actually implemented and used as a language generation component in a digital storytelling system is PRINCE, which is used as a front-end to the Proto-Propp plot generation system (Gervás et al. 2005). Language generation in this system is based on templates and schema's; a distinguishing feature is its capacity to generate analogies (Hervás et al. 2006).

In this paper we present another system for narrative generation: the Narrator, the NLG component of the Virtual Storyteller story generation system. We discuss its architecture and give an overview of how the different NLG tasks are carried out. Then we discuss two example stories generated by the Narrator, followed by some concluding remarks and pointers to future work. First, we briefly describe the Virtual Storyteller, the storytelling system of which the Narrator is a part.

9.2 The Virtual Storyteller

The Virtual Storyteller[2] is a multi-agent system that automatically creates fairy tales. Story generation in the Virtual Storyteller takes place in three stages, each handled by specialized agents.

The first stage is *plot generation*, which is based on the actions of semi-autonomous character agents in a simulated story world. These agents can reason logically and make plans to achieve their personal goals. In reaction to events and objects, they can experience emotions such as joy and distress, love and hate, and their subsequent actions are influenced by these emotions (Theune et al. 2004). Note that this is a so-called 'emergent narrative' approach (Aylett 1999) where stories are created by the characters, not based on a pre-authored plot or a story grammar.

During plot generation, a formal representation of what happens in the story world is constructed, called the Fabula (Swartjes and Theune 2006). When all events in the Story World have played out, the Fabula is passed on to the next stage: *narration*. This part of the story creation process is carried out by the Narrator agent, which maps the Fabula to a Dutch text using knowledge about discourse structure and Dutch syntax and morphology. In the rest of this paper, the workings of the Narrator will be discussed in some detail. The third and last stage is *presentation*: an embodied agent representing a human storyteller presents the narrative to the audience using text-to-speech. Our work on the generation of speech with a storytelling speech style is described in Theune et al. (2006b) and will not be discussed here.

[2] `http://wwwhome.cs.utwente.nl/~theune/VS/index.html`

9.3 The Narrator architecture

The design of the Narrator is based on the pipe-lined NLG architecture described by Reiter and Dale (2000), who distinguish three stages in the NLG process:

1. **Document planning**: determining what is to be said, and creating an abstract document specifying the structure of the information to be presented.
2. **Microplanning**: fleshing out the document specification by the generation of referring expressions, lexicalisation (word choice), and aggregation.
3. **Realisation**: converting the abstract document specification to real text, using knowledge about syntax, morphology, etc. In addition, mark-up may be added for use by external components.

Figure 9.1 shows the global architecture of the Narrator. It has three modules, corresponding to the three NLG stages described above: a Document Planner, a Microplanner and a Surface Realizer. The Document Planner receives a Fabula as input and turns it into a Document Plan, consisting of plot elements linked by rhetorical relations. The Microplanner converts the Document Plan into a so-called *Rhetorical Dependency Graph* by mapping the plot elements to partially lexicalised Dependency Trees. Finally, the Surface Realizer performs syntactic aggregation and the generation of referring expressions, and also takes care of linearization, morphology and punctuation to produce a proper surface form.

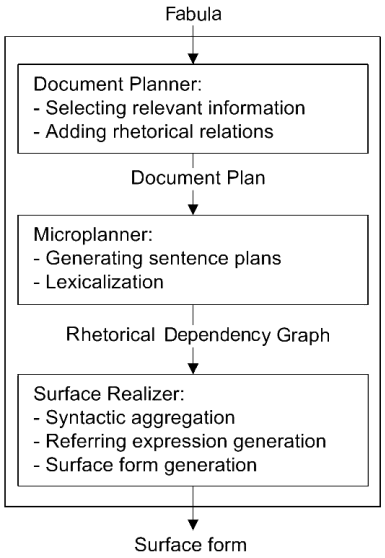


Figure 9.1: Architecture of the Narrator

The architecture of the Narrator deviates from the 'standard' NLG architecture of Reiter and Dale (2000) in that we situate syntactic aggregation in the Surface Realizer, whereas Reiter and Dale see aggregation as a Microplanning task. Cahill and Reape (1999) investigated the architecture of over twenty NLG systems and found that the location of the aggregation process varied widely across these systems. This divergence is partly caused by the fact that many, quite different processes are gathered under aggregation (Reape and Mellish 1999). However, in the Narrator we only focus on syntactic aggregation, which deals with grammatical processes and therefore in our view should be situated in the Surface Realizer. A consequence of this decision is that the generation of referring expressions is also located in the Surface Realizer: it would not be efficient to generate referring expressions that are at risk of later being deleted during ellipsis (which is part of the aggregation process). More importantly, to generate pronouns, the exact position of their antecedents has to be known.

9.4 Document Planning

The input for the Document Planning stage of the Narrator is a Fabula (Swartjes and Theune 2006): a causal network representing the story that emerged from the actions of the character agents in the story world. The Fabula does not form a complete network of everything that happened in the course of the story, but captures only those elements that have either a cause or an effect. Our model of Fabula structure is an adapted version of the story comprehension model of Trabasso et al. (1989). It has been implemented as an OWL ontology[3] and includes the following plot elements: actions, events, perceptions, goals, outcomes of goals, and characters' 'internal elements' such as emotions and beliefs. The possible relations between these plot elements are motivation, enablement, mental and physical cause relations. Also, each plot element is associated with a time stamp (in terms of discrete, virtual time steps in the story world) from which temporal relations between elements can be derived.

The Document Planner receives a Fabula as input and turns it into a Document Plan by mapping the causal links to appropriate rhetorical relations, removing irrelevant information and adding background information when necessary. We will illustrate this using the (simplified) example Fabula given in Figure 9.2. This Fabula represents a simple story about a dwarf who is hungry and believes there is an apple in the house. Combined, these two internal elements give rise to the goal to eat the apple. To achieve this goal, the dwarf carries out a simple plan: to take the apple and then eat it. Eating the apple leads consecutively to the perception and the belief that the apple has been eaten, which means a positive outcome for the original goal. Because the Fabula contains several elements that are relevant for plot generation but not for narration, the first step of the Document Planner is to prune away this irrelevant information. A typical example of this is the perception-belief-positive outcome chain following the action of eating the apple in the example Fabula: for the narration it is sufficient to mention only that the action was

[3] `http://www.w3.org/TR/owl-features/`

carried out, leaving it to the reader to infer the rest. Negative outcomes, however, are never pruned as these are generally quite relevant for the story.

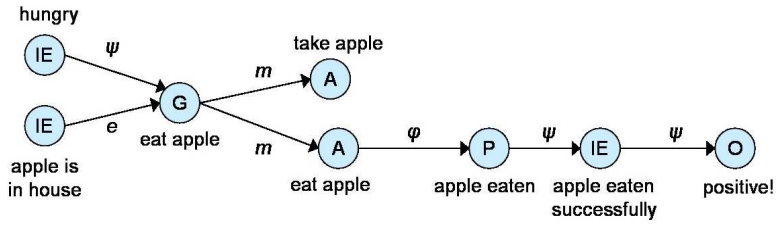


Figure 9.2: Example Fabula. (IE = internal element, G = goal, A = action, P = perception, O = outcome, ψ = physical cause, e = enablement, m = motivation, ϕ = psychological cause)

The next step is to convert the pruned Fabula to a binary tree and to replace the causal links with appropriate rhetorical relations, inspired by Rhetorical Structure Theory (RST) (Mann and Thompson 1987). The basic set of rhetorical relations used in the Narrator are Cause, Contrast, Temporal and Additive relations, with more specific relations such as Purpose and Elaboration as their subclasses.[4] When mapping the relations in the Fabula to rhetorical relations, consecutive steps of a plan are connected using a Temporal relation; motivation and psychological cause relations are mapped to Volitional Cause relations, and enablement and physical cause relations are mapped to Non-volitional Cause relations. Additive is the most general relation. It is used if two plot elements together cause another plot element, and more in general to connect two plot elements in the Document Plan if no more specific relation holds between them. The automatic assignment of Contrast relations is a subject of ongoing research.

The final step is to extend the Document Plan with information that is relevant for Narration, but which is not specified in the Fabula. Examples are information about the setting (introducing characters and locations) and information on the properties of characters and objects. In Figure 9.3, which shows the Document Plan corresponding to the Fabula from Figure 9.2, such added elements are shown in grey: a Setting element introducing the protagonist, connected via an Elaboration relation with an element specifying the protagonist's name. These added plot elements stand in a 'Temporal-once' (*Once upon a time...*) relation to the other elements; this particular relation was added specifically for the fairy tale domain.[5]

[4]Penning and Theune (2007) show that almost all cue words found in fairy tales fit into these classes.
[5]As pointed out by one of our reviewers, it might be more appropriate to classify this relation as Background. However, since it is used only for this one construction, its exact classification is somewhat academic.

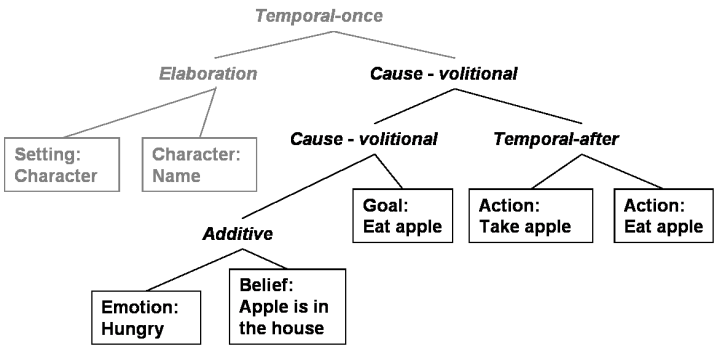


Figure 9.3: Document Plan based on the example Fabula from Figure 9.2.

9.5 Microplanning

The Microplanner maps the plot elements in the Document Plan to partially lexicalised Dependency Trees. We call the result a *Rhetorical Dependency Graph*: a graph (or rather, tree) structure with Dependency Trees expressing simple propositions as leaves, and rhetorical relations connecting them as nodes. Dependency Trees are an attractive formalism for use in the Narrator, in particular for the purpose of aggregation and ellipsis (see Section 9.6), because of the independence of word order, and the dependency labels that specify which role a node performs in its parent syntactic category.

In order to convert plot elements to Dependency Trees, sentence templates have been created that specify exactly how the arguments of the plot element should appear in the Dependency Tree. In total, over 30 different templates are currently available to express actions, events, failed actions, settings, states, beliefs and perceptions. Actions and events are expressed using a straightforward active voice construction, with an optional PP argument to express instruments, e.g., *De ridder opende de poort (met een sleutel)* (The knight opened the gate (with a key)). Failed actions are expressed using a complex sentence with *proberen* (to try) as the main verb, e.g., *De ridder probeerde de poort te openen* (The knight tried to open the gate). For internal states, we have standard constructions such as *De prinses was bang* (The princess was scared) and *De kabouter had honger* (The dwarf was hungry). In addition, templates are available for two specific storytelling-style constructions that allow for the expression of high-intensity emotions: sentences such as *Wat was ze gelukkig!* (Oh, how happy she was!) and *Ze was nog nooit zo gelukkig geweest!* (She had never been so happy before!). Such information concerning characters' emotions is usually included as an Elaboration relation in the Document Plan. Another specific storytelling construction is used for the setting: *Er was eens...* (Once upon a time, there was...). Various templates are available for different goal types such as Attain goals, where the agens wants to perform some

action or achieve some state (*Hij wilde de appel opeten / gelukkig zijn / de appel hebben*) (He wanted to eat the apple / be happy / have the apple) and Sustain goals, where the agens wants to maintain some existing situation (*Hij wilde blijven eten / gelukkig blijven / de appel houden*) (He wanted to keep eating / remain happy / keep the apple).

Once the sentence templates are selected, the trees are partially lexicalised. References to entities are not lexicalised, as this is part of the generation of referring expressions, which is done at a later stage. All other concepts are mapped to Dutch words by the lexical choice algorithm, which makes use of a discourse history to achieve some variation in wording, taking into account which words have been used recently.[6] The words added to the Dependency Trees are still uninflected, as morphology is taken care of during Surface Realization.

9.6 Aggregation

To achieve coherent output texts that are more than a sequence of simple sentences, syntactic aggregation is applied to the trees in the Rhetorical Dependency Graph. The aggregation algorithm goes through the graph depth-first, trying to combine the Dependency Trees at the leaf nodes. If aggregation succeeds, the graph is updated with a new, complex Dependency Tree replacing the original relation, and the algorithm continues looking for relations to transform.

The syntactic aggregation process consists of three steps. First, based on the rhetorical relation between two Dependency Trees, an appropriate cue word is selected that expresses this relation. Then, depending on the properties of the selected cue word, the two Dependency Trees may be joined together using a specific syntactic construction. Finally, the joined Dependency Trees are checked for repeated elements that can be ellipted. In the remainder of this section we briefly outline these steps; a detailed description of the aggregation process is given in Theune et al. (2006a).

For the purpose of cue word selection, a small taxonomy charting only the most prevalent cue words in Dutch has been constructed, using a variant of the substitutability test described by Knott and Dale (1994). The cue words are divided into four main classes, signaling Cause, Temporal, Contrast and Additive relations. Each of these classes is subdivided into more specific subclasses. A cue word from a subclass can always be replaced by a more general cue word in the same category. We have insufficient space to show the taxonomy here, but the original taxonomy (with 38 cue words) is given in Theune et al. (2006a), and an updated version (with 32 cue words) is presented in Penning and Theune (2007).

The rhetorical relation between two Dependency Trees in the Rhetorical Dependency Graph determines which cue words (if any) can be used to aggregate the trees. If the relation has no specific features licensing the use of a specialized cue word, a more general cue word is chosen. It is not necessarily the most specific applicable cue word that gets selected; discourse history plays a part as well. If a

[6] We use a small lexicon that was constructed specifically for our story domain and contains only a few synonyms; for a more sophisticated approach to lexical choice using WordNet, see Hervás et al. (2006).

cue word has been recently used, it is less likely to get chosen again. The selected cue word determines the structure of the generated sentence(s). If the cue word is a coordinator, a paratactic structure is created, i.e., a construction where two clauses of equal status are coordinated. A new Dependency Tree is constructed with a root labeled 'CONJ' (conjunction). Its child nodes are a coordinator (the cue word) and two conjuncts (the Dependency Trees to be aggregated). If the selected cue word is a subordinator, a hypotactic structure is created. If the cue word is an adverb, the cue word is added to either the first or the second tree in the relation (depending on the cue word), while the trees remain separate.

In the final step, ellipsis, superfluous nodes or branches are removed from an aggregated Dependency Tree. This only applies to paratactic trees, not to hypotactic ones where one of the combined clauses is subordinated to the other. First the identical nodes (if any) in the aggregated Dependency Tree are marked. We use unique identifiers to distinguish different instances of the same concept, so that ellipsis is only applied to nodes with identical referents. When all identical nodes (if any) have been found and marked, it is determined which operations are suitable, for example Conjunction Reduction, where the subject of the second clause is deleted. This operation is illustrated in Figure 9.4, expressing the Additive relation in the Document Plan of Figure 9.3. A corresponding surface string would be something like *De kabouter had honger en dacht dat er een appel in huis was* (The dwarf was hungry and believed there was an apple in the house). The other available forms of ellipsis are Gapping (deleting the main verb of the second clause, e.g., *De prinses at een appel en de kabouter een peer*) (The princess ate an apple and the dwarf a pear), Right Node Raising (deleting the rightmost string of the first clause, e.g., *De prinses ziet en de prins hoort de kabouter*) (The princess sees and the prince hears the dwarf), Stripping (deleting all constituents but one from the second clause, and replacing them by the word *ook* (too), as in *De prinses houdt van appels en de prins ook*) (The princess loves apples and so does the prince)[7] and Constituent Coordination (combining two non-identical constituents into one and deleting the rest of the second conjunct in its entirety, e.g., *De prins en de prinses houden van appels*) (The prince and the princess love apples).

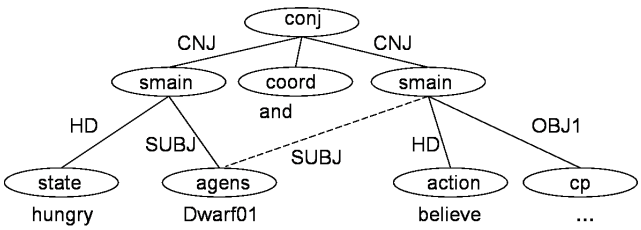


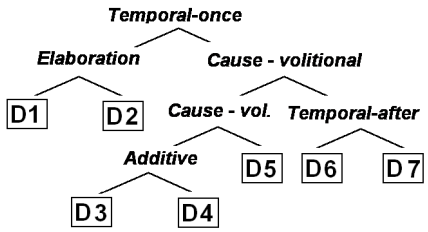
Figure 9.4: Dependency Tree with Conjunction Reduction.

The aggregation process is recursive in that an aggregated Dependency Tree

[7]Lit.: The princess loves apples and the prince too.

can potentially be combined with another tree it stands in a relation to. However, to keep the resulting sentences from getting too complex, we added the restriction that at most three simple Dependency Trees can be combined. In cases where this restriction prohibits aggregation it is still possible to express the relation between two Dependency Trees by adding an adverb such as *vervolgens* (then) or *echter* (however) to the second tree. To express the maximum of relations, after the initial traversal and transformation of the Rhetorical Dependency Graph, the algorithm makes another pass through it and expresses some final relations by adding adverbs to non-aggregated sentences.

We now illustrate the aggregation process using our example of the hungry dwarf. Figure 9.5 shows the Rhetorical Dependency Graph corresponding to the Document Plan from Figure 9.3, i.e., the same structure but with its leaves replaced by Dependency Trees (here abbreviated using a number). First, the two leftmost Dependency Trees D1 and D2 are combined. They are related by an Elaboration relation, so D2 is attached to D1 as a relative clause. Since the resulting Dependency Tree has a Temporal-once relation node as its parent, the cue phrase *Er was eens* (Once upon a time) is also added to the tree. Next, D3 and D4 are combined, resulting in the tree shown in Figure 9.4. D5 does not have a sister it can be combined with, so it is skipped in this pass through the tree. (Remember that the algorithm moves from left to right through the Rhetorical Dependency Graph, so D5 cannot be combined with its left sister at this point.) D6 and D7 are related by a temporal relation, so they are combined into a hypotactic structure starting with the cue phrase *nadat* (after). In a next pass through the Rhetorical Dependency Graph, the algorithm adds an adjunct to D5: the cue phrase *daarom* (therefore), which expresses the causal relation of the aggregated Dependency Tree from Figure 9.4 (D3 and D4) to D5. An overview of the result is shown to the right in Figure 9.5; the full text of the story is given in Section 9.9.



Er was eens D1 die D2
D3 en D4.
Daarom D5.
Nadat D6, D7.

Once upon a time, D1 who D2.
D3 and D4. Therefore D5.
After D6, D7.

Figure 9.5: Rhetorical Dependency Graph for the example story.

9.7 The generation of referring expressions

To decide whether a pronoun can be used to refer to a certain entity, or if it would be better to use a noun, we use an algorithm that combines and extends those of McCoy and Strube (1999) and Henschel et al. (2000). The algorithm is shown

in Figure 9.6. Its input is the referent r for which a referring expression is to be generated. It returns true if a pronoun should be used and false otherwise. Sometimes, even when a pronoun can be used without ambiguity, it is preferable to use a noun phrase for variation. An analysis of human-written fairy tales led us to a number of conclusions about when a noun phrase is preferred over a pronoun:

- At the beginning of a paragraph.
- If the antecedent has not been mentioned for two sentences.
- If a pronoun has been used a number of times (about four times) and the referring expression is the first one in the sentence.

Also, it is undesirable to use a pronoun when the referring expression should include additional information (e.g., information about the emotional state of a character). This information should be expressed by an adjective or a relative clause, which cannot be combined with a pronoun. If the above conditions do not hold, the algorithm returns true if there is strong parallelism with the previous clause or sentence (Chambers and Smyth 1998) or if the clause in which r appears stands in a Causal relation to the preceding clause (Kehler 2002). Otherwise the algorithm bases its decision on the salience of the referent, which is computed using the salience factors of Lappin and Leass (1994).

Pronominalize(r)

```
if first reference to r in current paragraph
  or antecedent has not been mentioned for two sentences
  or first reference in sentence and a pronoun has been used 4 times
  or referring expression should contain a relative clause
  or adjective should be added (determined by the Document Planner) then
  return false
end if
if r has not been mentioned in current sentence then
  if strong parallelism with previous sentence then
    return true
  end if
else
  if strong parallelism with first clause
    or r appears in causal relation then
    return true
  end if
end if
if r has highest salience value then
  return true
end if
return false
```

Figure 9.6: Algorithm used for pronominalization choice.

If a noun phrase is to be generated, the first step is to decide whether the name of the entity should be used or not (assuming the entity has a name). This decision is made randomly; 25% of the generated references use the name and the other 75% use a description. If the algorithm decides to generate a referring expression containing the entity's name, there are still two possibilities: simply the name (e.g., *Amalia*), or a noun phrase containing the name (*prinses Amalia*) (princess Amalia). The latter construction can only be used when the noun describes a function, such as princess, king or knight. If this is the case the algorithm includes the noun, otherwise it will only generate the name.

If a regular noun phrase is used instead of a name, first a noun has to be selected. To have some variation in the generated texts, for some concepts we have stored some synonyms in the lexicon: a preferred entry (the most commonly used word for that concept) plus one or more additional entries that will only be used occasionally. An example is the concept 'king' with the Dutch word *koning* as the preferred entry and the word *vorst* as an additional entry, which will only be used when the word *koning* has been used a number of times in a row. In addition, for some concepts hypernyms are available that can be used for variation once in a while. For example, *De ridder sloeg de prinses. Het meisje huilde* (The knight hit the princess. The girl cried).

After having selected the noun, three types of adjectives can be added to it:

1. Distinguishing adjectives, which are necessary in order to create an unambiguous referring expression. These are selected using a slightly modified version of the algorithm proposed by Krahmer and Theune (2002). When introducing a new character all known properties of this character are added to the referring expression, because they can be used as distinguishing adjectives later in the story.
2. Adjectives describing a character's internal state.
3. Adjectives that only have a decorative function. These adjectives are only added if the object to be described has no specific properties except its basic type; for example gates and bridges. The Narrator agent maintains a list of adjectives that can be used to 'spice up' the description of such objects, returning cliché expressions such as *een zware poort* (a heavy gate).

The final step of the noun phrase generation algorithm is choosing a determiner and adding this to the noun phrase generated so far. To this end an entity history is maintained. When an entity is mentioned for the first time, an indefinite article is used, and when the entity has been mentioned before, a definite article is used.

The algorithm described above can also create noun phrases that express relations of the referent with other objects, such as *de poort van het kasteel* (the gate of the castle). For the description of the related object, the noun phrase algorithm is applied recursively. In some of these cases, however, the relation can be easily inferred and it would be more appropriate not to mention it explicitly. For example, when the castle has already been mentioned, just saying *de poort* (the gate) is sufficient. Also, in some of these cases a definite article can be used for a first

mention, since the entity in question (e.g., the gate) has already been evoked by the mention of the related object (the castle), based on world knowledge ('every castle has a gate'). Such referring expressions are called *bridging descriptions*. To be able to generate this kind of description we have defined a number of inference rules such as $\forall x.Castle(x) \rightarrow \exists y.Gate(y) \wedge Has(x,y)$, which are checked if a referent r is related to another referent r' that has been mentioned earlier. So if r is a gate and r' is a castle that has been mentioned before, the algorithm then checks if there is a rule specifying that an entity of the type of r' usually has an entity of the type of r. If this is the case, then it checks if there is another salient entity that can also have an entity of the same type as r (so it checks if there is another entity that can have a gate – note that this can be another castle, but also an entity of a completely different type). Finally it checks if the entity r' has exactly one r, in which case a definite article can be used; if this is not the case an indefinite article will be used. A similar strategy is used for references to unique entities in the story; for example, in stories it is common to refer to a king as *the king* if there is only one king in the story. Such definite descriptions can be generated by checking if the Story World only contains one entity of this type.

9.8 Surface form generation

After aggregation and referring expression generation have taken place, the Surface Realiser linearises the Dependency Trees. It traverses the trees depth-first, ordering the children of each node by grammar rules that use the syntactic category of the parent node and the dependency labels of the child nodes. For example, the rule: SMAIN $\rightarrow$ SU + HD + OBJ1 states that if a parent node has syntactic category 'SMAIN' (sentence) and three children with dependency labels 'SU' (subject), 'OBJ1' (direct object) and 'HD' (main verb), then those children should be ordered in the above way. This particular rule would for instance be applied to produce the sentence *De prins zag Amalia* (The prince saw Amalia). Any nouns, adjectives and verbs are inflected at the moment they are linearised. Punctuation is added once linearisation is complete.

This concludes our description of the language generation process in the Narrator; more details can be found in Slabbers (2006).

9.9 Some example stories

After referring expression generation and surface realization have been applied, our simple example story about the hungry dwarf is finally narrated as follows:

> *Er was eens een kabouter die Plop heette. Hij had honger en dacht dat er een appel in een huis was. Daarom wilde hij de appel eten. Nadat Plop de appel had opgepakt, at hij de appel.*[8]

[8]Once upon a time there was a dwarf who was called Plop. He was hungry and believed there was an apple in a house. Therefore he wanted to eat the apple. After Plop had taken the apple, he ate it.

Note that the Referring Expression algorithm generates the indefinite noun phrase *een huis* (a house) instead of the bridging description *het huis* (the house), which would have been more appropriate if the house in question was Plop's house (which seems a reasonable assumption). However, in this case the Narrator lacked knowledge about the owner of the house and therefore produced a general description. Apart from this error, the output story is well-formed and coherent. But it is also very simple, and therefore we also show a more sophisticated example, generated from a hand-made Document Plan (shown in Figure 9.7). This input Document Plan contains Contrast relations and paragraph boundaries that cannot currently be generated automatically by the Document Planner, so this example illustrates the output level that could be achieved by the Narrator (in particular, the Microplanning and Surface Realisation components) once these remaining Document Planning problems are resolved.

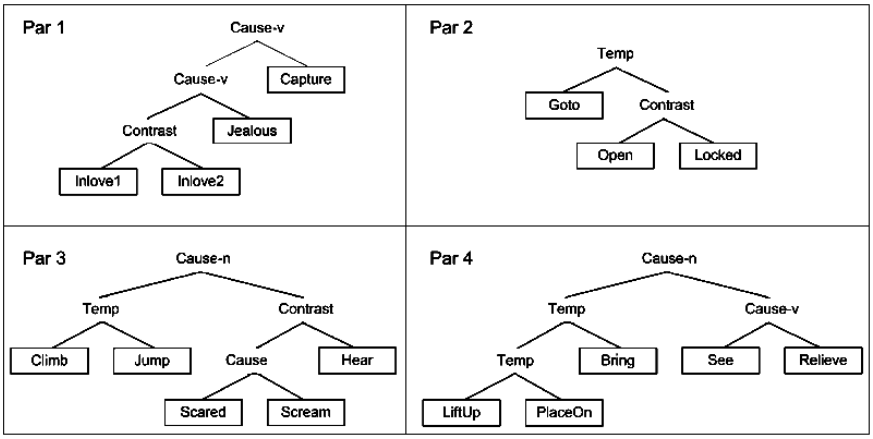


Figure 9.7: Initial Document Plan for the second example story.

Er was eens een mooie prinses, die Amalia heette. Een ridder van een ver land was verliefd op haar, maar zij was verliefd op een jonge prins. De ridder was jaloers, dus hij wilde haar ontvoeren.

De prinses woonde in een groot kasteel. Op een nacht ging de ridder naar het kasteel. Hij probeerde de zware poort te openen, maar die was op slot.

Nadat de ridder in een hoge boom was geklommen, sprong hij de slaapkamer van de prinses binnen. Zij was zo geschrokken, dat zij hard schreeuwde, maar niemand hoorde haar.

De ridder pakte de prinses op en vervolgens zette hij haar op zijn paard. Daarna bracht hij haar naar een oude en smalle brug. Aan

de overkant zag zij de prins, op wie zij verliefd was. Wat was prinses Amalia opgelucht![9]

This example story illustrates most of the NLG tasks described above, such as the addition of background information to the Document Plan (at the start of the first and second paragraphs), choice of cue words and aggregation, pronominalization and the expression of 'decorative' properties (*een groot kasteel, een hoge boom*) (a big castle; a high tree) and the use of specific storytelling constructions.

9.10 Conclusions and future work

In this paper we have presented the Narrator, a natural language generation component designed for use in a digital storytelling system, the Virtual Storyteller. The Narrator has been implemented (in Java), but it has only been tested with handmade input structures, because parts of the Document Planner and of the Virtual Storyteller's plot generation component are still under construction. So far, the only evaluations have been informal comparisons with the output of earlier versions of the Narrator.

The Narrator shows that the pipeline NLG architecture of Reiter and Dale (2000) can very well be used for the generation of narratives. It employs sophisticated algorithms for NLG tasks such as aggregation and the generation of referring expressions, enabling it to generate well-formed and fluent texts. This stands in contrast to the output of most digital storytelling systems, which usually consists of a straightforward mapping of plot elements to fixed expressions.

Unlike the STORYBOOK system (Callaway 2000), the Narrator cannot handle typical properties of narrative prose such as multiple viewpoints or character dialogue, and neither does it employ the type of narratological knowledge as the narrative generation architecture proposed by Lönneker (2005). However, it is capable of generating several linguistic constructions that are typical for fairy tale-like stories, and some narrative generation tasks are currently being investigated. These include the automatic placement of paragraph boundaries, detection of contrast relations and the lexical expression of emotions (taking the intensity of the emotion into account). Also, we would like to extend the Narrator so that it can also generate narratives in English. Since most algorithms and representations used in the Narrator are language independent, we expect that this should be relatively easy to accomplish by replacing the lexicon and the syntactic and morphological rules used for surface form generation.

Our main long-term challenge is to generate texts that are not only grammatical and coherent, but that can also really affect the reader by employing narrative

[9]Once upon a time there was a beautiful princess who was called Amalia. A knight from a far away country was in love with her, but she was in love with a young prince. The knight was jealous, so he wanted to abduct her. <P> The princess lived in a big castle. One night the knight went to the castle. He tried to open the gate, but it was locked. <P> After the knight had climbed a high tree, he jumped into the princess' bedroom. She was so scared that she screamed loudly, but nobody heard her. <P> The knight grabbed the princess and then he placed her on his horse. After that he took her to an old and narrow bridge. On the other side she saw the prince whom she was in love with. Oh, how relieved princess Amalia was!

techniques such as the use of subjective perspectives to heighten identification, and foreshadowing to increase suspense. Ablation tests in the style of Callaway (2000) could then be used to evaluate the effect of such techniques.

References

Aylett, R.(1999), Narrative in virtual environments – towards emergent narrative, *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*, pp. 83–86.

Cahill, L. and Reape, M.(1999), Component tasks in applied NLG systems, *Technical Report ITRI-99-05*, Information Technology Research Institute, ITRI, Brighton, UK.

Callaway, C.(2000), *Narrative Prose Generation*, PhD thesis, North Carolina State University, Raleigh, NC.

Chambers, G. and Smyth, R.(1998), Structural parallelism and discourse coherence: A test of Centering Theory, *Journal of Memory and Language* **39**, 593–608.

Gervás, P., Díaz-Agudo, B., Peinado, F. and Hervás, R.(2005), Story plot generation based on CBR, *Knowledge-Based Systems* **18**(4-5), 235–242.

Henschel, R., Cheng, H. and Poesio, M.(2000), Pronominalization revisited, *Proceedings of COLING*, pp. 306–312.

Hervás, R., Pereira, F., Gervás, P. and Cardoso, A.(2006), Cross-domain analogy in automated text generation, *Proceedings of the Third joint workshop on Computational Creativity, ECAI'06*, Trento, Italy.

Kehler, A.(2002), *Coherence, Reference, and the Theory of Grammar*, CSLI Publications.

Knott, A. and Dale, R.(1994), Using linguistic phenomena to motivate a set of coherence relations, *Discourse Processes* **18**(1), 35–62.

Krahmer, E. and Theune, M.(2002), Efficient context-sensitive generation of referring expressions, *in* K. van Deemter and R. Kibble (eds), *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, CSLI Publications, pp. 223–264.

Lappin, S. and Leass, H.(1994), An algorithm for pronominal anaphora resolution, *Computational Linguistics* **20**(4), 535–561.

Lönneker, B.(2005), Narratological knowledge for natural language generation, *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, Aberdeen, Scotland, pp. 91–100.

Mann, W. and Thompson, S.(1987), Rhetorical structure theory: A theory of text organization, *Technical Report ISI/RS-87-190*, ISI: Information Sciences Institute, Los Angeles, USA.

McCoy, K. and Strube, M.(1999), Generating anaphoric expressions: Pronoun or definite description?, *Proceedings of the ACL Workshop on The Relation of Discourse/Dialogue Structure and Reference*, pp. 63–71.

Penning, M. and Theune, M.(2007), Cueing the virtual storyteller: Analysis of cue phrase usage in fairy tales, *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG'07)*.

Reape, M. and Mellish, C.(1999), Just what is aggregation anyway?, *Proceedings of the 7th European Workshop on Natural Language Generation (ENLG'99)*, pp. 20–29.

Reiter, E. and Dale, R.(2000), *Building Natural Language Generation Systems*, Cambridge University Press, Cambridge.

Slabbers, N.(2006), *Narration for virtual storytelling*, Master's thesis, University of Twente.

Swartjes, I. and Theune, M.(2006), A Fabula model for emergent narrative, *Technologies for Interactive Digital Storytelling and Entertainment (TIDSE)*, Lecture Notes in Computer Science 4326, Springer-Verlag, pp. 95–100.

Theune, M., Hielkema, F. and Hendriks, P.(2006a), Performing aggregation and ellipsis using discourse structures, *Research on Language and Computation* **4**(4), 353–375.

Theune, M., Meijs, K., Heylen, D. and Ordelman, R.(2006b), Generating expressive speech for storytelling applications, *IEEE Transactions on Audio, Speech and Language Processing* **14**(4), 1137–1144.

Theune, M., Rensen, S., Op den Akker, R., Heylen, D. and Nijholt, A.(2004), Emotional characters for automatic plot creation, *in* S. Göbel and et al. (eds), *Technologies for Interactive Digital Storytelling and Entertainment (TIDSE)*, Lecture Notes in Computer Science 3105, Springer-Verlag, pp. 95–100.

Trabasso, T., Van den Broek, P. and Suh, S. Y.(1989), Logical necessity and transitivity of causal relations in stories, *Discourse Processes* **12**, 1–25.

10

Improved Sentence Alignment for Building a Parallel Subtitle Corpus

Building a Multilingual Parallel Subtitle Corpus

Jörg Tiedemann
University of Groningen

Abstract

In this paper on-going work of creating an extensive multilingual parallel corpus of movie subtitles is presented. The corpus currently contains roughly 23,000 pairs of aligned subtitles covering about 2,700 movies in 29 languages. Subtitles mainly consist of transcribed speech, sometimes in a very condensed way. Insertions, deletions and paraphrases are very frequent which makes them a challenging data set to work with especially when applying automatic sentence alignment. Standard alignment approaches rely on translation consistency either in terms of length or term translations or a combination of both. In the paper, we show that these approaches are not applicable for subtitles and we propose a new alignment approach based on time overlaps specifically designed for subtitles. In our experiments we obtain a significant improvement of alignment accuracy compared to standard length-based approaches.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

10.1 Introduction

The value of parallel corpora has been shown in various NLP applications and research disciplines. Some of them are data-driven machine translation (Brown et al. 1993, Brown 1996), multilingual lexicon/terminology extraction (Gale and Church 1991, Smadja et al. 1996, Hiemstra 1998, Gaussier 1998, Tufis and Barbu 2001), word sense disambiguation (Ide 2000, Diab and Resnik 2002) and general translation studies (Johansson 2002) to mention just a few. However, in contrast to monolingual language corpora there are still only a few parallel corpora available especially ones containing more than two languages. Often they originate from specialized domains such as legislation and administration or technical documentation and cover only a few "high density" languages. On the other hand, the amount of translated documents is increasing on the Internet even for lower density languages. In the past years several projects working on the collection of multilingual material from the web have been reported (Resnik 1999, Tiedemann and Nygard 2004).

One of the fastest growing multilingual resources are on-line databases of movie subtitles. There is a huge demand for subtitles on the Internet and users provide them to others in various languages via download services on-line. They are available in form of plain text files for modern as well as for classical movies and they are usually tagged with extra information such as language, genre, release year, user ratings and download counts. Subtitles are different to other parallel resources in various aspects: Most of them are (at least close to) transcriptions of spontaneous speech. They include plenty of idiomatic expressions and slang. They can easily be divided into different genres and time periods. There are even different subtitles versions (in the same language) for the same movie. Translations are usually very free and show a lot of cultural differences. They are aligned to the original movie and can therefore be linked to the actual sound signals. However, subtitles often summarize spoken utterances instead of completely transcribing them. Hence, they can also be used to study text compression and summarization (Daelemans et al. 2004). To summarize the discussion, subtitle databases provide a unique multilingual resource with various kinds of valuable information encoded in the texts.

In the following we will concentrate on building a parallel subtitles corpus from one on-line resource. In particular, we obtained the entire database of about 308,000 files from `http://www.opensubtitles.org`, a free on-line collection of movie subtitles in many languages. We are very grateful for the support by the providers of this website.

The paper is focused on the alignment of sentences and sentence fragments which is an essential step for building a parallel corpus. However, in the next section we first discuss necessary pre-processing steps to clean up the original database and to convert subtitle files into XML-based corpus files. Thereafter, we describe the details of the sentence alignment approaches in detail applied to our data. Finally, we present evaluations of the automatic alignment and provide some conclusions and prospects for future work.

10.2 Pre-processing

Pre-processing the original subtitle files is necessary because of several reasons: First of all, the database entirely consists of user uploads and, therefore, the content is not as clean as we want it to be. For example, movies are sometimes not tagged with the correct language, they are encoded in various character encodings, and they come in various formats. In our corpus we require a consistent format in a uniform encoding. In particular, we decided to use a simple standalone XML format and Unicode UTF-8. A sample output after all steps including tokenization is shown on the right-hand side of figure 10.1. Pre-processing consists of the following steps:

Subtitle format detection & conversion: We accepted two popular subtitle formats: SubRip files (usually with extension '.srt') and microDVD subtitle files (usually with extension '.sub'). For the conversion to XML we relied on the SubRip format which is more frequently used in the database we got. An example is shown in the left part of figure 10.1. microDVD subtitle files were converted to SubRip format using a freely available script sub2srt (`http://www.robelix.com/sub2srt/`).

Removing doubles: The database contains a lot of repeated subtitles; i.e. created for the same movie in the same language. We simply took the first one in the database and dismissed all the others. In future, we like to investigate possible improvements by other selection principles, e.g. taking download counts or user ratings into account.

Character encoding: All files are converted to Unicode UTF-8 to have a uniform encoding throughout all data files. This is especially useful when working with aligned data where several languages have to be put together. Unfortunately, we are not aware of a reliable classifier for automatic detection of character encodings and, therefore, we manually defined an incomplete encoding conversion table after inspecting sample data in various languages (see table 10.1).

Certainly, using a fixed language encoding table is only an ad-hoc solution causing errors in the conversion. However, using the language filter described below we remove most of the subtitles for which the encoding conversion failed. This, at least ensures high quality in our data as a trade-off for some quantity. In the future we would like to use an automatic classifier for better encoding detection of individual files.

Language Checking While processing the data we realized that many uploads are not correct and, for instance, contain text in a language different to the one specified. In order to filter them out we used an automatic classifier to check the language before accepting a subtitle file. For this we used `textcat` a freely available and trainable classifier designed for language identification (van Noord 2006).

```
00:00:26,500 --> 00:00:28,434
Spend all day with us.
00:00:28,502 --> 00:00:30,436
There are two--
pardon me--
00:00:30,504 --> 00:00:34,440
two of everything in
every Noah's arcade.
00:00:34,508 --> 00:00:36,361
That means
two of Zantar,
00:00:36,361 --> 00:00:36,884
That means
two of Zantar,
00:00:36,962 --> 00:00:40,454
Bay Wolf, Ninja Commando,
Snake-azon,
00:00:40,532 --> 00:00:41,464
Psycho Chopper...
00:00:41,533 --> 00:00:43,467
It's really good
seeing you, Benjamin.
```

```
<?xml version="1.0" encoding="utf-8"?>
<document>
  <s id="1">
    <time id="T1S" value="00:00:26,500" />
    <w id="1.1">Spend</w>
    <w id="1.2">all</w>
    <w id="1.3">day</w>
    <w id="1.4">with</w>
    <w id="1.5">us</w>
    <w id="1.6">.</w>
    <time id="T1E" value="00:00:28,434" />
  </s>
  <s id="2">
    <time id="T2S" value="00:00:28,502" />
    <w id="2.1">There</w>
    <w id="2.2">are</w>
    <w id="2.3">two</w>
    <w id="2.4">--</w>
    <w id="2.5">pardon</w>
    <w id="2.6">me</w>
    <w id="2.7">--</w>
    <time id="T2E" value="00:00:30,436" />
    <time id="T3S" value="00:00:30,504" />
    <w id="2.8">two</w>
    <w id="2.9">of</w>
    <w id="2.10">everything</w>
    <w id="2.11">in</w>
    <w id="2.12">every</w>
    <w id="2.13">Noah'</w>
    <w id="2.14">s</w>
    <w id="2.15">arcade</w>
    <w id="2.16">.</w>
    <time id="T3E" value="00:00:34,440" />
  </s>
```

Figure 10.1: A short segment of English subtitles of the movie "Wayne's World" from 1993 in SubRip (.srt) format (left) and a tokenized XML version of the first two sentences (right).

It uses N-gram models trained on example texts and, therefore, relies on the given encoding used in the training data. We applied the language checker after encoding conversion and, therefore, built language models for UTF-8 texts. For simplicity we used the training data from the `textcat` package converted to UTF-8 by means of the free Unix tool `recode`. Altogether, we created 46 language models. The classifier predicts for each given input file the most likely language according to the known models. The output of `textcat` is one of the following: (1) a certain classification of one language, (2) a ranked list of likely languages (in cases where the decision is not clear-cut), and, (3) a "resign" message in cases where the language classifier does not find any language that matches sufficiently enough. We accepted subtitles only in the case where the language classifier is certain that the language is the same as specified in the database.

Tokenization and sentence splitting: We used simple regular expressions for tokenization and sentence splitting. Tokenization of subtitles is a challenging task. First of all, there are various languages in the corpus and both, tokenization and sentence splitting are highly language dependent. However, for most languages

Table 10.1: Character Encoding Table

encoding	languages (ISO-639 codes)
cp1250	alb, bos, cze, pol, rum, scc, scr, slv, hrv
cp1251	bul, mac, rus
cp1252	afr, bre, cat, dan, dut, epo, est, fin, fre, ger, hun, ita, nor, pob, pol, por, spa, swe
cp1253	ell, gre
cp1254	tur
cp1255	heb
cp1256	ara
cp1257	lat, lit
iso-8859-4	ice
big5-eten	chi
shiftjis	jpn
euc-kr	kor

tokenizers and sentence boundary detectors are not readily available. We opted for a general solution using patterns in terms of regular expressions. For this we used Unicode character classes hoping to cover various languages equally well. The following patterns where defined:

- split between a non-punctuation character and a punctuation that is followed by either space, another punctuation or end-of-string.
- split between a punctuation and a non-punctuation character if they are preceeded by either start-of-string, another punctuation symbol or a white-space character.
- split punctuation symbols if they are not identical (leaving, for example '...' intact)
- split on all white-space characters

Note, that subtitles may contain HTML-like tags for formatting issues (like <i> and <font ...>). These tags have to be treated in a special way to avoid their tokenization.

Sentence boundary detection is also done with general patterns due to the lack of available tools for all languages involved. An issue specific to our data is the fact that subtitles contain many sentence fragments instead of well-formed grammatical sentences. Hence, even more sophisticated sentence splitters available for some languages will fail in many cases.

Table 10.2 shows the basic patterns used for detecting sentence boundaries. For Chinese, Korean and Japanese we simply split at standard punctuation symbols ".!? :" which works to some extent but, of course, is by far not an optimal solution for these languages. Note that sentences may span over several screens and may also stop in the middle of a screen. Figure 10.1 shows an example sentence (with

id=2) that spans over two screens. Hence, the patterns above are applied in a sliding window running through the data.

Table 10.2: Sentence splitting patterns using Perl regular expressions

pattern 1: split between “sentence end” and “sentence start”

sentence end:
`([^.]\.|[!?:])[\'\"]?(\s*|\Z)`
(a dot following a non-dot OR one of the following punctuation symbols “!? :” possibly followed by single or double quotes and white-space characters)
sentence start:
`(\A|\s+)\-?\s*[\"\’]?[\p{N}\p{Ps}\p{Lu}]`
(one or more white-spaces possibly followed by a hyphen, possibly followed by white-space characters and quotes followed by either a number, an opening punctuation or an uppercase letter.)

pattern 2: split between “sentence end” and “sentence start”

sentence end:
`[.!?:][\"\’\]\}\)]?\-?(\s*|\Z)`
(one of “.!? :” possibly followed by quotes, a hyphen and white-spaces)
sentence start:
`(\A|\s+)[\"\’]?[\¿\¡\p{Lu}]`
(one or more spaces, possibly followed by quotes and either an inverted initial question/exclamation mark or an uppercase letter)

The reason for applying two separate patterns is to combine different types of evidence when making decisions about sentence boundaries. Pattern 1 has stronger end-of-sentence constraints (hyphens and multiple dots are not allowed) in combination with a slightly relaxed sentence-start constraint (allowing for example digits and opening brackets) whereas pattern 2 has stronger sentence-start constraints (no digits and opening brackets) but relaxed sentence-end constraints (allowing hyphens and some closing brackets).

Shortcomings of our simple pattern based approach are obvious but they work reasonably well for many subtitle files and languages. However, both, tokenization and sentence splitting have to be improved in future work. It is impossible to find a general solution especially if we want to keep the collection as open as possible in terms of languages and genres. One general cue for sentence splitting might come from the time tags. Long pauses between subtitles may help to determine sentence boundaries. This is (only) one reason why we like to keep the time information in our corpus[1].

[1]Note that we decided to encode time slots as two separate “time events”, one for the starting time and one for the end. In this way we can handle sentences and time slots that cross each other which would otherwise not be possible to encode in XML.

Selection for alignment: Of course, not all movies are covered in all languages. Furthermore, there are several versions of movies around (for instance various video encodings, movie splits, etc) and, hence, several versions of subtitles fitting specific movie files. In order to yield the highest alignment quality, we selected only those subtitles that have been produced for exactly the same physical movie file.

The original database we obtained contains 232,643 subtitles for 18,900 movies in 59 languages. After pre-processing and filtering as described above we were left with 38,825 subtitle files in 29 languages. From that we selected 22,794 pairs of subtitles for alignment covering 2,780 movies in 361 language pairs. Altogether, this corresponds to about 22 million sentence alignments.

10.3 Subtitle alignment

Essential for building a parallel corpus is the alignment at some segmentation level. At least two segmentation approaches are possible for our data: alignment of subtitle screens (time slot segmentation) and alignment of sentences (using the sentence boundaries detected in the pre-processing phase). We decided to use the latter for three reasons: First, sentence alignment is a well established task that usually yields high accuracy with language independent methods. Secondly, sentences are linguistically motivated units and, therefore, more suitable for further processing than subtitle fragments shown together on screen. Very often these fragments are not coherent units; for example they may come from various speakers in one scene. Finally, the format of subtitles is very different in various languages due to visibility constraints and cultural differences. There will be lots of partial overlaps when comparing the contents of subtitle screens across different languages. This makes it more difficult to align these units.

There are many challenges when aligning subtitle sentences as illustrated in figure 10.2.

Subtitles often contain summarized information instead of literal transcriptions or translations. Hence, we can observe a lot of insertions, deletions and paraphrases when comparing various translations. Furthermore, sentence splitting introduce errors that make it difficult to solve certain alignment problems. For example, in figure 10.2, the first three Dutch subtitle screens are marked as one sentence although the first one actually corresponds to the movie title that should not be connected to the following sentences (and which is not included in the English version of the subtitles). Furthermore, there are untranslated fragments such as the third and the sixth screen in English which are embedded in other sentences. However, sentences are treated as units and, therefore, the only solution is to align such fragments together with the surrounding ones even though they do not have corresponding fragments in the other language. From this little example it becomes obvious that we cannot expect the same quality of standard sentence alignment approaches as reported in the literature for other text types. Nevertheless, it is interesting to see how far we can get with standard approaches and how we can improve them for our purposes.

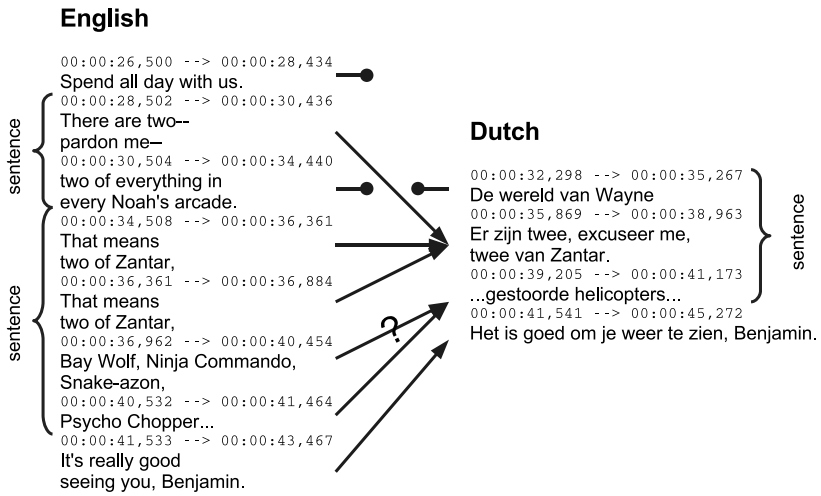


Figure 10.2: Alignment challenges: An example with English and Dutch subtitles.

10.3.1 Length-Based Sentence Alignment

There are several standard approaches to sentence alignment among them the popular length-based approach proposed by Gale and Church (1993). It is based on the assumption that translations tend to be of similar lengths (possibly factorized by a specific constant) with some variance. Using this assumption we can apply a dynamic algorithm to find the best alignment between sentences in one language and sentences in the other language. Another necessary assumption is that there are no crossing alignments (i.e. alignment is monotonic). Furthermore, alignments are restricted to certain types (usually 1:1, 1:0, 0:1, 2:1, 1:2 and 2:2) with prior probabilities attached to each type to make the algorithm more efficient and more accurate. In the default settings, there is a strong preference for 1:1 sentence alignments whereas the likelihood of the other types is very low. These settings are based on empirical studies of some example data (Gale and Church 1993).

It has been shown that this algorithm is very flexible and robust even without changing its parameters (Tjong Kim Sang 1996, Tiedemann and Nygard 2004). However, looking at our data it is obvious that certain settings and assumptions of the algorithm are not appropriate. As discussed above, we can observe many insertions and deletions in subtitle pairs and typically, a length-based approach cannot deal with such cases very well. Even worse, such insertions and deletions may cause a lot of follow-up errors due to the dynamic algorithm trying to cover the entire text in both languages. Nevertheless, we applied this standard approach with its default settings to our data to study its performance. Figure 10.3 shows an example of the length-based alignment approach.

As the figure illustrates there are many erroneous alignments using the stan-

English	Dutch
Spend all day with us . **There are two – pardon me – two of everything in every Noah' s arcade .**	*De wereld van Wayne* **Er zijn twee , excuseer me ,** *twee van Zantar . ... gestoorde helicopters ...*
That means two of Zantar , That means two of Zantar , Bay Wolf , Ninja Commando , Snakeazon , Psycho Chopper ...	*Het is goed om je weer te zien , Benjamin .*
It' s really good seeing you , Benjamin .	*Je bent al heel lang niet meer in Shakey' s geweest .*
You haven' t been into Shakey' s for so long .	*Ik heb het heel erg druk .*
Well , I' ve been real busy . **It' s two for you ' cause one won' t do .**	**Het zijn er twee voor jou , want eentje zal het niet doen .**
All this week , kids under 6 get every fifth – *There' s a new pet .*	**De hele week , krijgen kinderen onder de zes elke vijfde ...**
Ch- Ch- Chia Chia Pet – *the pottery that grows .*	*Er is een nieuw huisdier* **Het Chia huisdier .**
They are very fast . Simple .	*Het aardewerk dat groeit .*
Plug it in , and insert the plug from just about anything .	*Zij zijn erg snel .*
Simple .	*Simpel .*
Even for our customers in Waukegan , Elgin , and Aurora – We' ll be there right on time .	*Plug het in .*

Figure 10.3: Length-based sentence alignment - text in italics is wrongly aligned.

dard length-based approach. In fact, most of the alignments are wrong (in italics) and we can also see the typical problem of follow-up errors in this example[2]. An obvious idea to improve the alignment quality is to optimize the parameters of the original alignment approach. For example, we might want to change the prior probabilities of alignment types supported by the algorithm. Furthermore, we might also want to include other alignment types that frequently occur in the data. However, such a tuning would have to be done for each language pair and even within one language pair it is questionable if these parameters will be consistent to a large degree.

A second idea is to use the time information given in the subtitle files. As we have discussed before, there are many insertions and deletions in various subtitles and therefore the correspondence between source and target language is often not so obvious in terms of sentence lengths. However, subtitles usually span the entire movie and, therefore, they cover more or less the same amount of time. Assuming that corresponding text segments are shown roughly at the same time we can use the *time length* of each slot (screen) instead of sentence length to match source and target language sentences. Here we have to interpolate between "time events" in cases where sentences do not start or end at time slot boundaries. For this, we used simple linear interpolation between the two nearest time events. Now, the same algorithm using dynamic programming can be used only with lengths in time instead of lengths in characters. Figure 10.4 shows the result of our example

[2]Note, that 'Simpel' in the end of the example is aligned to the wrong instance of 'Simple' in English.

movie when using time lengths for alignment.

English	Dutch
Spend all day with us . **There are two – pardon me – two of everything in every Noah' s arcade .**	*De wereld van Wayne* **Er zijn twee , excuseer me ,** *twee van Zantar . ... gestoorde helicopters ...*
That means two of Zantar , That means two of Zantar , Bay Wolf , Ninja Commando , Snakeazon , Psycho Chopper ...	*Het is goed om je weer te zien , Benjamin .*
It' s really good seeing you , Benjamin .	*Je bent al heel lang niet meer in Shakey' s geweest .*
You haven' t been into Shakey' s for so long .	*Ik heb het heel erg druk .*
Well , I' ve been real busy .	*Het zijn er twee voor jou , want eentje zal het niet doen .*
It' s two for you ' cause one won' t do .	*De hele week , krijgen kinderen onder de zes elke vijfde ...*
All this week , kids under 6 get every fifth – **There' s a new pet .**	**Er is een nieuw huisdier** *Het Chia huisdier .*
Ch- Ch- Chia Chia Pet – **the pottery that grows .**	**Het aardewerk dat groeit .**
They are very fast .	**Zij zijn erg snel .**
Simple .	**Simpel .**
Plug it in , and insert the plug from just about anything .	**Plug het in .**

Figure 10.4: Sentence alignment based on time lengths - text in italics is wrongly aligned.

Unfortunately, the time length approach also produces a lot of errors. A striking difference is that in the end of the example the algorithm synchronizes well between source and target language which reduces the amount of follow-up errors from this point on. However, the accuracy is still unsatisfactory concluding from our first impressions. This will also be supported by our evaluations presented in section 10.4.

10.3.2 Alignment with Time Overlaps

As seen in the previous sections, length-based approaches cannot deal very well with our data collection. Let us now consider a different approach directly incorporating the time information given in the subtitles. The intuition in this approach is roughly the same as in the previous one based on time lengths: corresponding sentences are shown at roughly the same time because they should be synchronized with the movie. However, in the previous approach we only used the time length to match sentences but now we will directly use the absolute time values. Using start and end time for each sentence (using the same interpolation technique as before) we can measure the overlap between source and target language segments. We can now sequentially go through the subtitles and try to find segments with the highest overlap. This can be done efficiently in linear time without recursions because we use absolute times that cannot be shifted around. Again, we define a set of alignment types that we like to support in our alignment

Figure 10.5: Sentence alignment with time overlaps

program. In particular, we use 1:1, 1:0, 0:1, 2:1, 1:2, 3:1, 1:3, 1:4, and 4:1 alignments[3]. Using these settings we can check possible alignments at each position and its surroundings and select the one with the highest overlap before moving to the next positions. The general principle of time overlap alignment is illustrated in figure 10.5.

One of the big advantages of this approach is that it can easily handle insertions and deletions at any position as long as the timing is synchronized between the two subtitle files. Especially initial and final insertions often cause follow-up errors in length-based approaches but they do not cause any trouble in the time overlap approach (look for example at the first English sentence in the example in figure 10.5). Remaining errors mainly occur due to sentence splitting errors and timing differences. The latter will be discussed in the end of the following section. The result of the alignment with time overlaps for our example data is shown in figure 10.6.

10.4 Evaluation

In order to see the differences between the alignment approaches discussed above we manually evaluated a small sample of our aligned data. We selected two language pairs, Dutch-English and Dutch-German, and randomly selected five movies

[3]Note that the set of alignment types is different to the standard length-based sentence alignment approach. The impact of these additional types on the alignment quality has not been investigated.

English	Dutch
Spend all day with us .	
There are two – pardon me – two of everything in every Noah' s arcade . That means two of Zantar , That means two of Zantar , Bay Wolf , Ninja Commando , Snake- azon , Psycho Chopper ...	*De wereld van Wayne* **Er zijn twee , excuseer me , twee van Zantar . ... gestoorde helicopters ...**
It' s really good seeing you , Benjamin .	**Het is goed om je weer te zien , Benjamin .**
You haven' t been into Shakey' s for so long .	**Je bent al heel lang niet meer in Shakey' s geweest .**
Well , I' ve been real busy .	**Ik heb het heel erg druk .**
It' s two for you ' cause one won' t do .	**Het zijn er twee voor jou , want eentje zal het niet doen .**
All this week , kids under 6 get every fifth – There' s a new pet .	**De hele week , krijgen kinderen onder de zes elke vijfde ... Er is een nieuw huisdier** *Het Chia huisdier .*
Ch- Ch- Chia Chia Pet – **the pottery that grows .**	**Het aardewerk dat groeit .**
They are very fast .	**Zij zijn erg snel .**
Simple .	
Simpel . **Plug it in , and insert the plug from just about anything .**	**Plug het in .** *Het is simple !*

Figure 10.6: Sentence alignment based on time overlaps - text in italics is wrongly aligned.

for each of them[4]. In order to account for differences in alignment quality at different positions we selected 10 initial, 10 final sentence alignments, and 10 alignments in the middle of each document for each of the three alignment approaches. The evaluation was carried out by one human expert using the following three grades: correct, partially correct and wrong.

The overall result of our evaluation is shown in table 10.3[5].

As expected, the length-based approaches are much less accurate than the time-overlap approach. Surprisingly, the scores for alignments based on time lengths performs even worse than the standard sentence length based approach. The fact that Dutch-German performs much better for the time overlap approach should not be seen as a general tendency. The difference is due to the selection of movies which is different for the two language pairs. This is illustrated in table 10.4 showing the detailed scores per movie and language pair using the time overlap approach.

We can see that there are three movies that perform very poorly with our time-overlap approach, two in Dutch-English and one in Dutch-German. This explains the difference in total scores when comparing the two language pairs. The scores

[4]Each sub-corpus contains different movie pairs according to the subtitles available for the particular language pair. We did not want to restrict the selection to movies that have subtitles in all three languages. Hence, we obtained different sets of movies for both language pairs for our evaluation.

[5]We omit details over alignment positions. There are no striking differences in accuracies between initial, final and intermediate alignments with one exception: Time-length alignments performed much worse for the final sentence alignments than for the other ones. The reason for this is unclear to the author.

Table 10.3: Evaluation of alignment accuracy per alignment approach

alignment type	languages	correct	partially	wrong
sentence length	dut-eng	64.2%	9.2%	26.6%
sentence length	dut-ger	62.3%	12.3%	25.3%
time length	dut-eng	54.6%	6.9%	38.6%
time length	dut-ger	57.5%	9.8%	32.7%
time overlap	dut-eng	73.1%	8.7%	18.2%
time overlap	dut-ger	85.7%	6.8%	7.5%

Table 10.4: Evaluation of accuracy of time-overlap alignment per movie

languages	movie	correct	partially	wrong
dut-eng	Cube Zero	22.4%	14.3%	63.3%
dut-eng	Finding Nemo	36.8%	18.4%	44.7%
dut-eng	Grizzly Man	84.1%	12.7%	3.2%
dut-eng	Training Day	96.8%	3.2%	0.0%
dut-eng	Win a Date with Tad Hamilton	100.0%	0.0%	0.0%
dut-ger	Batman	92.6%	0.0%	7.4%
dut-ger	Cidade de deus	100.0%	0.0%	0.0%
dut-ger	Peggy Sue got married	82.8%	10.3%	6.9%
dut-ger	Rush Hour 2	93.5%	6.5%	0.0%
dut-ger	The Ring	33.3%	26.7%	40.0%

also show that the time-overlap approach either works very well (around or above 90% correct) or very poorly (below 40% correct). Here we see a clear effect of timing differences. If the timing is (only slightly) different for the two subtitle files to be aligned, the performance of the time-overlap approach drops dramatically. That means, if both subtitles are not synchronized very well with each other, almost everything goes wrong using the time-overlap approach whereas length-based approaches are not effected by this. These timing differences appear due to two factors: (1) the *subtitle speed* might be different, and, (2) the *time offset* for starting the subtitles might be different.

The solution to the problem mentioned above is to adjust the time values in one of the subtitles to synchronize it with the other one. In other words, we have to find the parameters for time offset and speed difference. In fact, appropriate values can easily be computed using two fixed anchor points at different positions in the movie, preferably far away from each other. Using the current time values at these fix-points it is a matter of simple maths to calculate offset and speed difference (or time ratio) assuming that the speed is constant in both subtitles.

The difficulty now is to find such reference points. One way is to add them by hand. We developed a simple tool for interactive sentence alignment, ISA (Tiedemann 2006), which can be used for this task. The tool allows to add break points at any place in subtitle pairs to be used for offset and speed calculation. Table 10.5 shows the results after manually adding such fix-points (one in the beginning and one at the end of the movie) to the three problematic movie pairs and re-aligning them after synchronization. The evaluation is done in the same way as before (using 10 initial, 10 final and 10 intermediate sentence alignments).

Table 10.5: Sentence alignment with speed and time offset adjustments using manual fix-points (*time ratio* refers to the speed difference and *offset* is the time offset in seconds).

movie	time ratio	offset	correct	partially	wrong
Cube Zero	0.9997	2.378	76.2%	9.5%	14.3%
Finding Nemo	0.9996	0.470	84.1%	4.5%	11.4%
The Ring	0.9589	0.302	100.0%	0.0%	0.0%

All alignments have been improved significantly with only very little human intervention. The speed and offset parameters fixed most of the alignment errors. Remaining errors are often due to little shifts in displaying subtitles and could easily be fixed using ISA as well. It is interesting to see that both parameters are very close to their default values (1 for time ratio and 0 for offset) but still have a significant impact on the alignment quality. It shows how brittle the time-overlap alignment approach is with respect to subtitle synchronization.

Although the manual intervention helped to improve the alignment quality significantly it is not reasonable to run the alignment in this way on the entire corpus with its more than 22,000 subtitle pairs. However, simple heuristics could be used to detect pairs for which an inspection would be desirable. For example, subtitle alignments with surprisingly many empty alignments (1:0 or 0:1) are likely to contain errors. They could be selected and presented to users via the ISA interface for validation.

Another technique to add break points for synchronization is to look for cognates in source and target language subtitles. Subtitles can be scanned from the beginning and from the end to find appropriate cognates. Simple string matching techniques and fixed thresholds can be used to spot candidate pairs. Assuming that they appear at the same position in the movie they can be used for calculating the two parameters necessary. However, experiments have shown that using such a technique in general decreases the overall alignment performance significantly. This is due to false hits where cognates are found but they do not refer to true reference positions. A reason for this is that names are often repeated in various contexts but not in the same way in all translations. Hence, correct matches but at non-corresponding positions frequently occur. However, cognate based techniques can again be combined with the same heuristics presented above: using

this approach only in cases where the alignment seems to contain errors indicated by many empty alignments. This combined approach will be investigated in future work.

10.5 Conclusions

In this paper, a new multilingual parallel corpus consisting of movie subtitles in 29 languages has been presented. The corpus contains about 23,000 aligned subtitle pairs with altogether about 22 million sentence alignments. The data has been tokenized and sentences boundaries have been marked to be stored in a uniform XML format. We investigated three approaches to automatic sentence alignment, two based on length correspondence and a novel algorithm based on time overlaps. The latter yields significantly higher accuracies than traditional length-based alignment approaches. Remaining errors can be fixed to a large degree with minor human intervention. The corpus is available for research purposes and we will work on its extension in the future.

References

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D. and Mercer, R. L.(1993), The mathematics of statistcal machine translation: Parameter estimation, *Computational Linguistics* **19**(2), 263–311.

Brown, R. D.(1996), Example-based machine translation in the Pangloss system, *Proceedings of the 16th International Conference on Computational Linguistics, COLING-96*, Copenhagen, Denmark, pp. 169–174.

Daelemans, W., Höthker, A. and Tjong Kim Sang, E.(2004), Automatic sentence simplification for subtitling in Dutch and English, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004)*, Lisbon, Portugal.

Diab, M. and Resnik, P.(2002), An unsupervised method for word sense tagging using parallel corpora, *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia.

Gale, W. A. and Church, K. W.(1991), Identifying word correspondences in parallel texts, *Proceedings of the DARPA SNL Workshop*.

Gale, W. A. and Church, K. W.(1993), A program for aligning sentences in bilingual corpora, *Computational Linguistics* **19**, 75–102.

Gaussier, E.(1998), Flow network models for word alignment and terminology extraction from bilingual corpora, *Proceedings of COLING-ACL-98*, Montreal, pp. 444–450.

Hiemstra, D.(1998), Multilingual domain modeling in Twenty-One: Automatic creation of a bi-directional translation lexicon from a parallel corpus, *Proceedings of the eighth CLIN meeting*, pp. 41–58.

Ide, N.(2000), Cross-lingual sense determination: Can it work?, *Computers and the Humanities, Special Issue on the Proceedings of the*

SIGLEX/SENSEVAL Workshop, A. Kilgarriff and M. Palmer, eds. **34**(1-2), 223–34.

Johansson, S.(2002), Towards a multilingual corpus for contrastive analysis and translation studies, *in* L. Borin (ed.), *Parallel Corpora, Parallel Worlds*, number 43 in *Language and Computers: Studies in Practical Linguistics*, Rodopi, Amsterdam, New York, pp. 47–59.

Resnik, P.(1999), Mining the web for bilingual text, *37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, Maryland.

Smadja, F., McKeown, K. R. and Hatzivassiloglou, V.(1996), Translating collocations for bilingual lexicons: A statistical approach, *Computational Linguistics, 22(1)*.

Tiedemann, J.(2006), ISA & ICA - two web interfaces for interactive alignment of bitexts, *Proceedings of the 5th International Conference on Language Resources and Evaluation, (LREC'2006)*, Genova, Italy.

Tiedemann, J. and Nygard, L.(2004), The OPUS corpus - parallel and free, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004)*, Lisbon, Portugal.

Tjong Kim Sang, E. F.(1996), Aligning the Scania Corpus, *Technical report*, Department of Linguistics, University of Uppsala.

Tufis, D. and Barbu, A.-M.(2001), Automatic construction of translation lexicons, *Proceedings of the WSES and IEEE International Conference on Multimedia, Internet, Video Technologies*, Malta, pp. 2181–2186.

van Noord, G.(2006), TextCat, http://www.let.rug.nl/∼vannoord/TextCat/.

11

Automatic Extraction of Dutch Hypernym-Hyponym Pairs

Erik Tjong Kim Sang and Katja Hofmann
University of Amsterdam

Abstract

In this study, we apply pattern-based methods to text for extracting lexical data, in particular the hypernymy relation. We automatically derive thousands of interesting lexical patterns like *such NP as NP* and evaluate the performance of these patterns by comparing the information they extract from a newspaper corpus with the information in the Dutch part of EuroWordNet. Additionally we investigate the usefulness of combining hypernymy relation evidence generated by different patterns and compare this approach with the application of fixed patterns to web data. We find that with larger quantities of data, individual fixed extraction patterns outperform the large combination of patterns applied to the corpus.

11.1 Introduction

WordNet is a key lexical resource for natural language applications. However its coverage (currently 155k synsets for the English WordNet 2.0) is far from complete. For languages other than English, the available WordNets are considerably smaller, like for Dutch with a 44k synset WordNet. Here, the lack of coverage creates bigger problems. A manual extension of the WordNets is costly. Currently, there is a lot of interest in automatic techniques for updating and extending taxonomies like WordNet.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

Hearst (1992) was the first to apply fixed syntactic patterns like *such NP as NP* for extracting hypernym-hyponym pairs. Carballo (1999) built noun hierarchies from evidence collected from conjunctions. Pantel et al. (2004) learned syntactic patterns for identifying hypernym relations and combined these with clusters built from co-occurrence information. Pasca (2004) applied lexico-syntactic patterns for extracting labeled name categories from web data. Recently, Snow et al. (2005) generated tens of thousands of hypernym patterns and combined these with noun clusters to generate high-precision suggestions for unknown noun insertion into WordNet (Snow et al. 2006). All previously mentioned papers deal with English.

Little work has been done for Dutch. Van der Plas and Bouma (2005) employed noun distribution characteristics for extending the Dutch part of EuroWordNet with named entities and their definitions. IJzereef (2004) used fixed patterns to extract Dutch hypernyms from text and encyclopedias. In this paper we will extend this work in two ways. First, we will apply techniques which automatically derive extraction patterns for lexical relations from text corpora. Information for arbitrary relations can be derived in this way. We concentrate on the relation which is most useful for our own goal of extending the Dutch WordNet: hypernymy. Second, we apply the best extraction patterns of our corpus work to the largest available text resource: the web. We evaluate both approaches by comparing the information that they derive with the available WordNet.

In section two we introduce the task, hypernym extraction. Section three and four presents our text corpus work and our web extraction work[1], respectively. Section five concludes the paper.

11.2 Task and Approach

We examine techniques for automatically extending WordNets. In this section we describe which relation we focus on, explain some data preprocessing steps, describe the information we are looking for and introduce our evaluation approach.

11.2.1 Task

We concentrate on a particular semantic relation: hypernymy. One term is a hypernym of another if its meaning both covers the meaning of the second term and is broader. For example, *furniture* is a hypernym of *table*. The opposite term for hypernym is hyponym. So *table* is a hyponym of *furniture*. Hypernymy is a transitive relation. If term A is a hypernym of term B while term B is a hypernym of term C then term A is also a hypernym of term C.

In WordNet, hypernym relations are defined between senses of words (synsets). The Dutch WordNet (DWN), which is a part of EuroWordNet (Vossen 1998), contains 659,284 of such hypernym noun pairs of which 100,268 are immediate links and 559,016 are inherited by transitivity. More importantly, the resource contains hypernym information for 45,979 different nouns. A test with a recent Dutch newspaper text revealed that the Dutch WordNet only covered about two-thirds of the

[1] Results of the web experiments were earlier published in Tjong Kim Sang (2007).

noun lemmas in the newspaper (among the missing words were *e-mail*, *euro* and *provider*). Proper names, like names for persons, organizations and locations, pose an even larger problem: DWN only contains 1608 words that start with a capital character.

11.2.2 Natural language processing

We aim at developing extraction techniques which are fast and robust. Therefore we try to use as little natural language processing preprocessing as possible. In particular, we refrain from using full parsers because we expect them to lack the speed to handle large quantities of (web) data and because we expect them to fail when having to deal with incomplete sentences, like those in web snippets and tabular data.

However, completely skipping preprocessing is not feasible. In this study we apply the following preprocessing methods to the source texts:

- Tokenizing: separating punctuation marks from words and identifying sentence boundaries
- Part-of-speech tagging: assigning word classes to tokens
- Lemmatizing: assigning lemmas to tokens

We deliberately avoided using a parser in order to limit the required time and resources for processing the corpus. In a future study, we will compare the performances of our approach with different preprocessing strategies, one of which will be dependency parsing.

For the web queries, we also need to be able to determine plural versions of nouns. For this purpose we use the plural list from CELEX (Baayen et al. 1995) (64,040 nouns). Words that are not present in the database, receive a plural form which is determined by a machine learner trained on the database. It has the seven final characters of the words as features and can predict 152 different plural forms. Its leave-one-out accuracy on the training set is 89%.

11.2.3 Collecting evidence

We search the web for fixed patterns like *such H as A, B and C*. Following Snow et al. (2006) , we derive two types of evidence from these patterns:

- *H* is a hypernym of *A*, *B* and *C*
- *A*, *B* and *C* are siblings of each other

Here, *sibling* refers to the relative position of the words in the hypernymy tree. Two words are siblings of each other if they share a parent.

We compute a hypernym evidence score $s(h, w)$ for each candidate hypernym h for word w. It is the sum of the normalized evidence for the hypernymy relation

between h and w, and the evidence for sibling relations between w and known hyponyms c of h:

$$s(h,w) = \frac{f_{hw}}{\sum_x f_{xw}} + \sum_c \frac{g_{cw}}{\sum_y g_{yw}}$$

where f_{hw} is the frequency of patterns that predict that h is a hypernym of w, g_{cw} is the frequency of patterns that predict that c is a sibling of w, and x and y are arbitrary words from the WordNet. For each word w, we select the candidate hypernym h with the largest score $s(h,w)$.

For each hyponym, we only consider evidence for hypernyms and siblings. We have experimented with different scoring schemes, for example by including evidence from hypernyms of hypernyms and remote siblings, but found this basic scoring scheme to perform best.

11.2.4 Evaluation

We use the Dutch part of EuroWordNet (DWN) (Vossen 1998) for evaluation of our hypernym extraction methods. Hypernym-hyponym pairs that are present in the lexicon are assumed to be correct. In order for the evaluation to be complete, we also need negative examples, pairs of words that are not related by hypernymy. For this purpose, we make the same assumption as Snow et al. (2005) : the hypernymy relations in the WordNets are complete for the terms that they contain. This means that when two words are present in the lexicon without the target relation being specified between them, then we assume that the target relation does not hold between them. The presence of positive and negative relations allows for an automatic evaluation in which precision, recall and F values are computed.

We do not require our search method to find the exact position of a target word in DWN. Instead, we are satisfied with any ancestor. In order to rule out identification methods which simply return the top node of the hierarchy for all words, we also measure the distance between the assigned hypernym and the target word. The ideal distance is one which would occur if the ancestor is a parent. A grandparent receives distance two and so on.

We compare our work with two alternative methods for hypernym extraction found in the literature. The first is based on conjunctions: it considers the pattern *A, B and C* as evidence for the fact that *A*, *B* and *C* share a hypernym (Caraballo 1999). A disadvantage of this pattern is that the hypernym information it suggests, is indirect and more noisy than the best hypernym pattern. However, this pattern occurs frequently and allows for deriving more information.

The second alternative, we examine, is the hypernym extraction approach of Sabou et al. (2005): assume that the longest known character suffix of the hyponym is a hypernym. This morphological approach maps *blackbird* to *bird*. It is very useful for Dutch in which compounding nouns is the rule rather than an exception. As extra constraints for this method we require that the candidate hypernym should already be present in DWN and that the split point in the word

should be chosen in such a way that the word is split in two parts which both contain at least three characters.

11.3 Hypernym extraction from a text corpus

In this section we describe the hypernymy extraction work applied to a newspaper corpus. First, we evaluate a method for automatically deriving corpus-specific extraction patterns from a set of examples. After this we examine a method for combining these patterns and compare the performance of the combination with the best individual patterns and the morphological approach described in section 11.2.4.

11.3.1 Extracting individual patterns

In this study, we used the Twente Nieuws Corpus, a corpus of Dutch newspaper text and subtitle text covering four years (1999-2002) and containing about 300M words. The corpus was processed by automatic tools which tokenized it, assigned part-of-speech tags and identified lemmas. Next we used the same approach as Snow et al. (2005) but with lexical information rather than dependency parses: all pairs of nouns with four or fewer tokens (words or punctuation signs) between them were selected. The intermediate tokens (labeled `infix`) as well as the token before the first noun (`prefix`) and the token following the second noun (`suffix`) were stored as a pattern. For each noun pair, four patterns were identified:

- `N1 infix N2`
- `prefix N1 infix N2`
- `prefix N1 infix N2 suffix`
- `N1 infix N2 suffix`

The patterns also included information about whether the nouns were singular or plural, a feature which can be derived from the part-of-speech tags. We identified 3,283,492 unique patterns. The patterns were evaluated by registering how often they assigned correct hypernym relations correspond to noun pairs from DWN. Only 118,306 patterns had a recall that was larger than zero. The majority of these patterns (63%) had a precision of 1.0 but the recall of these patterns was very low (0.00003-0.00025). The highest registered recall value for a single pattern was 0.00897 (for *N-pl and N-pl*). The recall values are low because of the difficulty of the task: we aim at generating a valid hypernym for *all* 45,979 nouns in the Dutch WordNet. A recall value of 1.0 corresponds with single pattern predicting a correct hypernym for every noun in DWN, something which is impossible to achieve.

Table 11.1 lists ten top-precision patterns of the format `N1 infix N2` and a recall score of 0.0005 or higher. Figure 11.1 contains an overview of the precision and recall values of all 421 patterns of that group. For comparison with other approaches, we have selected the pattern *N zoals N*, a combination of the results

Precision	**Recall**	**F$_{\beta=1}$**	**Dist,**	**Pattern**
0.375	0.00137	0.00273	2.56	N-pl , vooral N-pl (*especially*)
0.300	0.00133	0.00264	2.23	N-pl , waaronder N-pl (*among which*)
0.258	0.00120	0.00238	1.55	N-pl , waaronder N-sg (*among which*)
0.250	0.00196	0.00388	2.08	N-pl of ander N-pl (*or other*)
0.244	0.00418	0.00821	1.96	N-pl zoals N-sg (*such as*)
0.220	0.00259	0.00512	2.10	N-pl zoals N-pl (*such as*)
0.213	0.00809	0.01559	1.99	N-pl en ander N-pl (*and other*)
0.205	0.00387	0.00760	2.20	N-pl , zoals N-pl (*such as*)
0.184	0.00396	0.00775	1.78	N-pl , zoals N-sg (*such as*)
0.158	0.00394	0.00768	1.68	N-sg en ander N-pl (*and other*)

Table 11.1: Top ten high precision patterns of the format `N1 infix N2` extracted from the text corpus which have a recall score higher than 0.00100. In the patterns, N-pl and N-sg represent a plural noun and a singular noun, respectively. It is possible to aggregate patterns by ignoring the number of the noun (N-pl + N-sg = N) in order to achieve higher recall scores at the expense of lower precision rates. The phrase between parentheses is an English translation of the main words of the pattern.

of four patterns of which two are listed in Table 11.1. This pattern obtained a precision score of 0.22 and a recall score of 0.0068 (Table 11.2).

11.3.2 Combining corpus patterns

Snow et al. (2005) showed that for the task of collecting hypernym-hyponym pairs, a combination of extraction patterns outperforms the best individual pattern. In order to obtain a combined prediction of a set of patterns, they represented word pairs by a sequence of numeric features. The value of each feature was determined by a single pattern predicting that the word pair was related according to the hypernymy relation or not. A machine learning method, Bayesian Logistic Regression was used to determine the combined prediction of feature sets for unknown word pairs based on a comparison with known word pairs which could be part of the relation or not.

We have replicated this work of Snow et al. (2005) for our Dutch data. We have identified 16728 features which corresponded with hypernym-hyponym extraction patterns. All noun pairs which were associated with at least five of these patterns in the text corpus, were represented by numerical features which encoded the fact that the corresponding pattern predicted that the two were related (value 1) or not (value 0). Only nouns present in the Dutch WordNet (DWN) were considered. The class associated with each feature set could either be positive if the ordered word pair occurred in the hypernymy relation of DWN or negative if the ordered pair was not in the DWN relation. This resulted in a dataset of 528,232 different ordered pairs of which 10,653 (2.0%) were related.

The performance of the combined patterns was determined by 10-fold cross

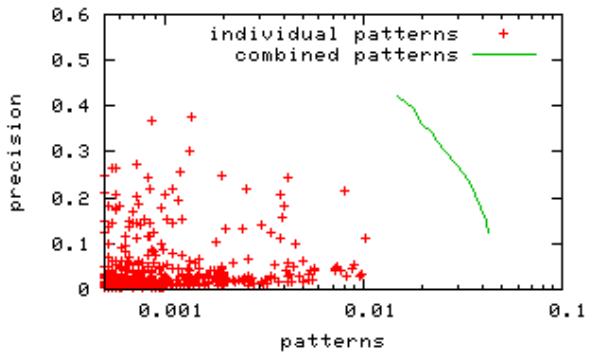


Figure 11.1: Precision and recall values of the 421 hypernym-hyponym extraction patterns of the format `N1 infix N2` with the highest recall values when applied to the text corpus (+) compared with combinations of these patterns (line). Pattern combinations outperform individual patterns both with respect to precision and recall. All recall values are low because of the difficulty of the task (reproducing valid hypernyms for all nouns in the WordNet).

validation: the training set was divided in ten parts and the classes for each part were predicted by using the other nine parts as training data. Like Snow et al. (2005), we used Bayesian Logistic Regression as learning technique (Genkin et al. 2004). We have also tested Support Vector Machines but these proved to be unable to process the data within a reasonable time.

The classifier assigned a confidence score between 0 and 1 to each pair. We computed precision and recall values for different acceptance threshold values (0.001-0.90) which resulted in the line in Figure 11.1. The combined patterns obtain similar precision scores as the best individual patterns but their recall scores are a lot higher. For comparison with other approaches, we have used acceptance threshold 0.5, which resulted in a precision of 0.36 and a recall of 0.020 (Table 11.2).

Surprisingly enough, both alternative hypernym prediction methods outperform the combination of lexical patterns (Table 11.2). The conjunctive pattern obtains a lower precision score than the combination but its recall is an order of magnitude larger than that of the combination. The morphological approach of selecting the shortest suffix that is also a valid word as the candidate hypernym (*blackbird* $\rightarrow$ *bird*), does even better: obtaining precision, recall and distance scores that are the best of all examined approaches. The morphological approach is limited in its application: it cannot find out that a *poodle* is a *dog* because the latter word is not part of the former. Therefore we need to look for another approach for finding more good hypernym-hyponym pairs.

Method	**Prec.**	**Recall**	**F**$_{\beta=1}$	**Dist.**
corpus: *N zoals N*	0.22	0.0068	0.013	2.01
corpus: combined	0.36	0.020	0.038	2.86
corpus: *N en N*	0.31	0.14	0.19	1.98
morphological approach	0.54	0.33	0.41	1.19

Table 11.2: Performances measured with the corpus approach and the morphological approach. The pattern combination perform better than the best individual pattern but both suffer from low recall figures. The conjunctive pattern and the morphological approach, predicting the longest known suffix of each word as its hypernym (section 11.2.4), surprisingly enough outperform both corpus approaches on most evaluation measures.

11.4 Extraction from the web

In this section we describe our web extraction work. First we discuss the format of the web queries. Then we present the results of the web extraction work and compare them with the results of the earlier described extraction from text corpora (section 11.3) and the morphological approach (section 11.2.4). We conclude with an analysis of the errors made by the best system.

11.4.1 Query format

In order to collect evidence for lexical relations, we search the web for lexical patterns. When working with a fixed corpus on disk, an exhaustive search can be performed. For web search, however, this is not possible. Instead, we rely on acquiring interesting lexical patterns from text snippets returned for specific queries. The format of the queries has been based on three considerations.

First, a general query like *such as* is insufficient for obtaining much interesting information. Most web search engines impose a limit on the number of results returned from a query (for example 1000), which limits the opportunities for assessing the performance of such a general pattern. In order to obtain useful information, the query needs to be more specific. For the pattern *such as*, we have two options: adding the hypernym, which gives *hypernym such as*, or adding the hyponym, which results in *such as hyponym*.

Both extensions of the general pattern have their disadvantages. A pattern that includes the hypernym may fail to generate much useful information if the hypernym has many hyponyms. And patterns with hyponyms require more queries than patterns with hypernyms (at least one per child rather than one per parent). We chose to include hyponyms in the patterns. This approach models the real-world task in which someone is looking for the meaning of an unknown entity.

The final consideration regards which hyponyms to use in the queries. Our focus is on evaluating the approach via comparison with an existing WordNet. Rather than flooding the search engine with queries representing every hyponym in the lexical resource, we chose to search only for a random sample of hypernyms.

We observed the evaluation score to converge for approximately 1500 words and this is the number of queries we settled for.

11.4.2 Web extraction results

For our web extraction work, we used two fixed context patterns: one containing the word *zoals* (*such as*), a reliable and reasonably frequent hypernym pattern according to our corpus work, and another containing the word *en* (*and*), the most frequent pattern found in the text corpus. We chose to add randomly selected candidate hyponyms to the queries to improve the chance to retrieve interesting information.

This approach worked well. As Table 11.3 shows, both patterns outperformed the F-rate of the combined patterns in the corpus experiments. Like in the corpus experiments, the conjunctive web pattern outperformed the *such as* web pattern with respect to precision and recall. We assume that the frequency of the two patterns plays an important role (the Google index contains about five time as many pages with the conjunctive pattern in comparison with pages with *zoals*).

Finally, we combined word-internal information with the conjunctive pattern approach by adding the morphological candidates to the web evidence before computing hypernym pair scores. This approach achieved the highest recall at only slight precision loss (Table 11.3). A basic combination approach by using the conjunctive pattern for searching for hypernyms for hyponyms for which no candidates were generated by the morphological approach, would have achieved a similar performance.

Method	**Prec.**	**Recall**	$\mathbf{F}_{\beta=1}$	**Dist.**
web: *N zoals N*	0.23	0.089	0.13	2.06
web: *N en N*	0.39	0.31	0.35	2.04
morphological approach	0.54	0.33	0.41	1.19
web: *en* + morphology	0.48	0.45	0.46	1.64

Table 11.3: Performances measured in the two web experiments and a combination of the best web approach with the morphological approach. The conjunctive web pattern *N en N* rates best, because of its high frequency. All evaluation rates can be improved by supplying the best web approach with word-internal information.

11.4.3 Error analysis

We have inspected the output of the conjunctive web extraction with word-internal information. For this purpose we have selected the ten most frequent hypernym pairs (top group, see Table 11.4), the ten least frequent (bottom group) and the ten pairs exactly between these two groups (center group). 40% of the pairs were correct, 47% incorrect and 13% were plausible but contained relations that were not present in the reference WordNet. In the center group all errors were caused

by the morphological approach while all other errors in the top group and in the bottom group originated from the web extraction method.

11.5 Concluding remarks

The contributions of this paper are two-fold. First, we show that the large quantity of available web data allows basic patterns to perform better on hypernym extraction than an advanced combination of extraction patterns applied to a large corpus. Second, we demonstrate that the performance web extraction can be improved by combining its results with those of a corpus-independent morphological approach.

While the web results are of reasonable quality, some concern can be expressed about the quality of the corpus results. At best, we obtained an F-value of 0.038 which is a lot lower than than the 0.348 reported for English in Snow et al. (2005). There are two reasons for this difference. First, the evaluation methods are different: we aim at generating hypernyms for all words in the WordNet while Snow et al. only look for hypernyms for words in the WordNet *that are present in their corpus*. Second, in their extraction work Snow et al. also use a sense-tagged corpus, a resource which is unavailable for Dutch.

One of the directions of future work will be to compare the lexical patterns applied in this paper to the dependency patterns like used by Snow et al. (2005). The first indications from this work are promising. If we interpret the results of Hofmann and Tjong Kim Sang (2007) with the evaluation methods used for creating Table 11.2, we obtain scores which are similar to the scores of the combined lexical patterns. Further experiments are necessary to check if these initial scores can be improved and if dependency patterns can be applied successively to web snippets.

The described approach has already been applied in a project for extending the coverage of the Dutch WordNet. However, we remain interested in obtaining better performance levels, especially in higher recall scores. There are some suggestions on how we could achieve this. First, our present selection method, which ignores all but the first hypernym suggestion, is quite strict. We expect that the lower-ranked hypernyms include a reasonable number of correct candidates as well. Second, a combination of web patterns could outperform individual patterns if we include the conjunctive pattern in the combination. Obtaining results for many different web pattens will be a challenge given the restrictions on the number of web queries we can currently use.

Acknowledgements

Both authors are supported by research projects funded by the Dutch Science Foundation (NWO). Katja Hofmann received a grant by the NWO project Cornetto. Erik Tjong Kim Sang received grants from both the Cornetto and the NWO project IMIX.

+/-	score	hyponym	hypernym
-	912	buffel	predator
+	762	trui	kledingstuk
?	715	motorfiets	motorrijtuig
+	697	kruidnagel	specerij
-	680	concours	samenzijn
+	676	koopwoning	woongelegenheid
+	672	inspecteur	opziener
?	660	roller	werktuig
?	654	rente	verdiensten
?	650	cluster	afd.

Table 11.4: Example output of the the conjunctive web system with word-internal information. Of the ten most frequent pairs, four are correct (+). Four others are plausible but are missing in the WordNet (?).

References

Baayen, R., Piepenbrock, R. and Gulikers, L.(1995), *The CELEX Lexical Database (Release 2) [CD-ROM]*, Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania.

Caraballo, S. A.(1999), Automatic construction of a hypernym-labeled noun hierarchy from text, *Proceedings of ACL-99*, Maryland, USA.

Genkin, A., Lewis, D. D. and Madigan, D.(2004), *Large-Scale Bayesian Logistic Regression for Text Categorization*, Technical report, Rutgers University, New Jersey.

Hearst, M. A.(1992), Automatic acquisition of hyponyms from large text corpora, *Proceedings of ACL-92*, Newark, Delaware, USA.

Hofmann, K. and Tjong Kim Sang, E.(2007), Automatic extension of non-english wordnets, *Proceedings of SIGIR'07*, Amsterdam, The Netherlands.

IJzereef, L.(2004), *Automatische extractie van hyperniemrelaties uit grote tekstcorpora*, MSc thesis, University of Groningen.

Pantel, P., Ravichandran, D. and Hovy, E.(2004), Towards terascale knowledge acquisition, *Proceedings of COLING 2004*, Geneva, Switzerland, pp. 771–777.

Pasca, M.(2004), Acquisition of categorized named entities for web search, *Proceedings of CIKM 2004*, Washington, USA.

Sabou, M., Wroe, C., Goble, C. and Mishne, G.(2005), Learning domain ontologies for web service descriptions: an experiment in bioinformatics, *14th International World Wide Web Conference (WWW2005)*, Chiba, Japan.

Snow, R., Jurafsky, D. and Ng, A. Y.(2005), Learning syntactic patterns for automatic hypernym discovery, *NIPS 2005*, Vancouver, Canada.

Snow, R., Jurafsky, D. and Ng, A. Y.(2006), Semantic taxonomy induction from

heterogenous evidence, *Proceedings of COLING/ACL 2006*, Sydney, Australia.

Tjong Kim Sang, E.(2007), Extracting hypernym pairs from the web, *Proceedings of ACL-2007*, Prague, Czech Republic.

Van der Plas, L. and Bouma, G.(2005), Automatic acquisition of lexico-semantic knowledge for qa, *Proceedings of the IJCNLP Workshop on Ontologies and Lexical Resources*, Jeju Island, Korea.

Vossen, P.(1998), *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publisher.

12

Lexico-Semantic Multiword Expression Extraction

Tim Van de Cruys and Begoña Villada Moirón
University of Groningen

Abstract

This paper describes a fully unsupervised and automated method for the large-scale extraction of multiword expressions (MWEs) from large corpora. The method takes into account the non-compositionality of MWEs; the intuition is that a noun within a MWE cannot easily be replaced by a semantically similar noun. To implement this intuition, a noun clustering is automatically extracted (using distributional similarity measures), which gives us clusters of semantically related nouns. Next, a number of statistical measures – based on selectional preferences – is developed that formalize the intuition of non-compositionality. The ratio of individual noun preference over cluster preference shows how likely a particular expression is to be a MWE (i.e. whether or not an individual noun accounts for all the preference of a certain cluster). Our approach has been tested on Dutch, and has been both manually and automatically evaluated.

12.1 Introduction

MWEs are expressions whose linguistic behaviour is not predictable from the linguistic behaviour of their component words. Baldwin (2006) characterizes the idiosyncratic behavior of MWEs as "a lack of compositionality manifest at different levels of analysis, namely, lexical, morphological, syntactic, semantic, pragmatic

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

and statistical". One property that seems to affect MWEs the most is semantic non-compositionality. MWEs are typically non-compositional. As a consequence, it is not possible to replace the content words of a MWE by semantically related words. Take for example the expressions in (1) and (2):

(1) a. break the vase
b. break the cup
c. break the dish

(2) a. break the ice
b. *break the snow
c. *break the hail

Expression (1) is fully compositional. Therefore, *vase* can easily be replaced with semantically related nouns such as *cup* and *dish*. Expression (2), on the contrary, is non-compositional; it is impossible to replace *ice* with semantically related words, such as *snow* and *hail*. Note that we assume a dual classification of expressions into compositional and non-compositional instances; we ignore the possibility that expressions fall in a continuum between compositionality and non-compositionality with many fuzzy cases in between. By 'fuzzy cases' we refer to expressions that are neither fully compositional nor fully non-compositional; such expressions may involve metaphoricity or figurative language.

Due to their non-compositionality, current proposals argue that MWEs need to be described in the lexicon (Sag et al. 2002). In most languages, electronic lexical resources (such as dictionaries, thesauri, ontologies) suffer from a limited coverage of MWEs. To facilitate the update and expansion of language resources, the NLP community would clearly benefit from automated methods that extract MWEs from large text collections. This is the main motivation to pursue an automated and fully unsupervised MWE extraction method.

12.2 Previous work

Recent proposals that attempt to capture semantic compositionality (or lack thereof) employ various strategies. Approaches evaluated so far make use of dictionaries with semantic annotation (Piao et al. 2006), WordNet (Pearce 2001), automatically generated thesauri (Lin 1999, Fazly and Stevenson 2006, McCarthy et al. 2003), vector-based methods that measure semantic distance (Baldwin et al. 2003, Katz and Giesbrecht 2006), translations extracted from parallel corpora (Villada Moirón and Tiedemann 2006) or hybrid methods that use machine learning techniques informed by features coded using some of the above methods (Venkatapathy and Joshi 2005).

Pearce (2001) describes a method to extract collocations from corpora by measuring semantic compositionality. The underlying assumption is that a fully compositional expression allows synonym replacement of its component words, whereas a collocation does not. Pearce measures to what degree a collocation candidate allows synonym replacement. The measurement is used to rank candidates

relative to their compositionality.

Building on Lin (1998), McCarthy et al. (2003) measure the semantic similarity between expressions (verb particles) as a whole and their component words (verb). They exploit contextual features and frequency information in order to assess meaning overlap. They established that human compositionality judgements correlate well with those measures that take into account the semantics of the particle. Contrary to these measures, multiword extraction statistics (log-likelihood, mutual information) poorly correlate with human judgements.

A different approach proposed by Villada Moirón and Tiedemann (2006) measures translational entropy as a sign of meaning predictability, and therefore non-compositionality. The entropy observed among word alignments of a potential MWE varies: highly predictable alignments show less entropy and probably correspond to compositional expressions. Data sparseness and polysemy pose problems because the translational entropy cannot be accurately calculated.

Fazly and Stevenson (2006) use lexical and syntactic fixedness as partial indicators of non-compositionality. Their method uses Lin's (1998) automatically generated thesaurus to compute a metric of lexical fixedness. Lexical fixedness measures the deviation between the pointwise mutual information of a verb-object phrase and the average pointwise mutual information of the expressions resulting from substituting the noun by its synonyms in the original phrase. This measure is similar to Lin's (1999) proposal for finding non-compositional phrases. Separately, a syntactic flexibility score measures the probability of seeing a candidate in a set of pre-selected syntactic patterns. The assumption is that non-compositional expressions score high in idiomaticity, that is, a score resulting from the combination of lexical fixedness and syntactic flexibility. The authors report an 80% accuracy in distinguishing literal from idiomatic expressions in a test set of 200 expressions. The performance of both metrics is stable across all frequency ranges.

In this study, we are interested in establishing whether a fully unsupervised method can capture the (partial or) non-compositionality of MWEs. The method should not depend on the existence of large (open domain) parallel corpora or sense tagged corpora. Also, the method should not require numerous adjustments when applied to new subclasses of MWEs, for instance, when coding empirical attributes of the candidates. Similar to Lin (1999), McCarthy et al. (2003) and Fazly and Stevenson (2006), our method makes use of automatically generated thesauri; the technique used to compile the thesauri differs from previous work. Aiming at finding a method of general applicability, the measures to capture non-compositionality differ from those employed in earlier work.

12.3 Methodology

In the description and evaluation of our algorithm, we focus on the extraction of verbal MWEs that contain prepositional complements, although the method could easily be generalized to other kinds of MWEs.

In our semantics-based approach, we want to formalize the intuition of non-compositionality, so that MWE extraction can be done in a fully automated way. A

number of statistical measures are developed that try to capture the MWE's non-compositional bond between a verb-preposition combination and its noun by comparing the particular noun of a MWE candidate to other semantically related nouns.

12.3.1 Data extraction

The MWE candidates (verb + prepositional phrase) are automatically extracted from the *Twente Nieuws Corpus* (Ordelman 2002), a large corpus of Dutch newspaper texts (500 million words), which has been automatically parsed by the Dutch dependency parser Alpino (van Noord 2006). Next, a matrix is created of the 5,000 most frequent verb-preposition combinations by the 10,000 most frequent nouns, containing the frequency of each MWE candidate.[1] To this matrix, a number of statistical measures are applied to determine the non-compositionality of the candidate MWEs. These statistical measures are explained in §12.3.3.

12.3.2 Clustering

In order to compare a noun to its semantically related nouns, a noun clustering is created. These clusters are automatically extracted using standard distributional similarity techniques (Weeds 2003, van der Plas and Bouma 2005). First, dependency triples are extracted from the *Twente Nieuws Corpus*. Next, feature vectors are created for each noun, containing the frequency of the dependency relations in which the noun occurs.[2] This way, a frequency matrix of 10K nouns by 100K dependency relations is constructed. The cell frequencies are replaced by pointwise mutual information scores (Church et al. 1991), so that more informative features get a higher weight. The noun vectors are then clustered into 1,000 clusters using a simple K-means clustering algorithm (MacQueen 1967) with cosine similarity. During development, several other clustering algorithms and parameters have been tested, but the settings described above gave us the best EuroWordNet similarity score (using Wu and Palmer (1994)).

Note that our clustering algorithm is a hard clustering algorithm, which means that a certain noun can only be assigned to one cluster. This may pose a problem for polysemous nouns. On the other hand, this makes the computation of our metrics straightforward, since we do not have to decide among various senses of a word. In future work, we want to investigate the use of soft clustering algorithms, that take into account the various senses of a noun.

12.3.3 Measures

The measures used to find MWEs are inspired by Resnik's method to find selectional preferences (Resnik 1993, Resnik 1996). Resnik uses a number of measures based on the Kullback-Leibler divergence, to measure the difference between the

[1] The lowest frequency verb-preposition combination (with regard to the 10,000 nouns) appears 3 times.
[2] E.g. dependency relations that qualify *apple* might be 'object of *eat*' and 'adjective *red*'. This gives us dependency triples like $< apple, obj, eat >$.

prior probability of a noun class $p(c)$ and the probability of the class given a verb $p(c|v)$. We adopt the method for particular nouns, and add a measure for determining the 'unique preference' of a noun given other nouns in the cluster, which, we claim, yields a measure of non-compositionality. In total, four measures are used, the latter two being the symmetric counterpart of the former two.

12.3.3.1 Verb preference

The first two measures, $A_{v \to n}$ (equation 12.2) and $R_{v \to n}$ (equation 12.3), formalize the unique preference of the verb[3] for the noun. Equation 12.1 gives the Kullback-Leibler divergence between the overall probability distribution of the nouns and the probability distribution of the nouns given a verb; it is used as a normalization constant in equation 12.2. Equation 12.2 models the actual preference of the verb for the noun.

$$S_v = \sum_n p(n \mid v)\ log \frac{p(n \mid v)}{p(n)} \tag{12.1}$$

$$A_{v \to n} = \frac{p(n \mid v)\ log \frac{p(n|v)}{p(n)}}{S_v} \tag{12.2}$$

When $p(n|v)$ is 0, $A_{v \to n}$ is undefined. In this case, we assign a score of 0.

Equation 12.3 gives the ratio of the verb preference for a particular noun, compared to the other nouns that are present in the cluster.

$$R_{v \to n} = \frac{A_{v \to n}}{\sum_{n' \epsilon C} A_{v \to n'}} \tag{12.3}$$

When $R_{v \to n}$ is more or less equally divided among the different nouns in the cluster, there is no preference of the verb for a particular noun in the cluster, whereas scores close to 1 indicate a 'unique' preference of the verb for a particular noun in the cluster. Candidates whose $R_{v \to n}$ value approaches 1 are likely to be non-compositional expressions.

12.3.3.2 Noun preference

In the latter two measures, $A_{n \to v}$ and $R_{n \to v}$, the direction of preference is changed: they model the unique preference of the noun for the verb. Equation 12.4 models the Kullback-Leibler divergence between the overall probability distribution of verbs, and the distribution of the verbs given a certain noun. It is used again as a normalization constant in equation 12.5, which models the preference of the noun for the verb.

$$S_n = \sum_v p(v \mid n)\ log \frac{p(v \mid n)}{p(v)} \tag{12.4}$$

[3] We will use 'verb' to designate a prepositional verb, i.e. a combination of a verb and a preposition.

$$A_{n \rightarrow v} = \frac{p(v \mid n)\ log \frac{p(v|n)}{p(v)}}{S_n} \tag{12.5}$$

When $p(v|n)$ is 0, $A_{n \rightarrow v}$ is undefined. In this case, we again assign a score of 0.

Equation 12.6 gives the ratio of noun preference for a particular verb, compared to the other nouns that are present in the cluster.

$$R_{n \rightarrow v} = \frac{A_{n \rightarrow v}}{\sum_{n' \epsilon C} A_{n' \rightarrow v}} \tag{12.6}$$

Both measures have the same characteristics as the ones that model verb preference. If a noun shows a much higher preference for a verb than the other nouns in the cluster, we expect that the candidate expression tends towards non-compositionality.

Note that the measures for verb preference and the measures for noun preference are different in nature. It is possible that a certain verb only selects a restricted set of nouns, while the nouns themselves can co-occur with many different verbs. This brings about different probability distributions. In our evaluation, we want to investigate the impact of both preferences.

12.3.3.3 Lexical fixedness measure

For reasons of comparison, we also evaluated the lexical fixedness measure – based on pointwise mutual information – proposed by Fazly and Stevenson (2006).[4] The lexical fixedness is computed following equation 12.7

$$Fixedness_{lex}(v, n) = \frac{PMI(v, n) - \overline{PMI}}{s} \tag{12.7}$$

where $\overline{PMI}$ stands for the mean given the cluster, and s for the standard deviation. Note that Fazly and Stevenson (2006) use the M most similar nouns given a certain noun, while we use all nouns in a cluster. This means that our M-value varies.

12.3.4 Example

In this section, an elaborated example is presented, to show how our method works. Take for example the two MWE candidates in (3):

(3) a. in de smaak vallen
in the taste fall
to be appreciated

[4] Fazly and Stevenson (2006) combine the lexical fixedness measure with a measure of syntactic flexibility. Here, we only compare our method to the former measure, concentrating on non-compositionality rather than syntactic rigidity.

b. in de put vallen
in the well fall
to fall down the well

In the first expression, *smaak* cannot be replaced with other semantically similar nouns, such as *geur* 'smell' and *zicht* 'sight', whereas in the second expression, *put* can easily be replaced with other semantically similar words, such as *kuil* 'hole' and *krater* 'crater'.

The first step in the formalization of this intuition, is the extraction of the clusters in which the words *smaak* and *put* appear from our clustering database. This gives us the clusters in (4).

(4) a. **smaak**: *aroma* 'aroma', *gehoor* 'hearing', *geur* 'smell', *gezichtsvermogen* 'sight', *reuk* 'smell', *spraak* 'speech', *zicht* 'sight'
b. **put**: *afgrond* 'abyss', *bouwput* 'building excavation', *gaatje* 'hole', *gat* 'hole', *hiaat* 'gap', *hol* 'cave', *kloof* 'gap', *krater* 'crater', *kuil* 'hole', *lacune* 'lacuna', *leemte* 'gap', *valkuil* 'pitfall'

Next, the various measures described in §12.3.3.1 and §12.3.3.2 are applied. Resulting scores are given in tables 12.1 and 12.2.

MWE candidate	$A_{v\to n}$	$R_{v\to n}$	$A_{n\to v}$	$R_{n\to v}$
val#in smaak	.12	1.00	.04	1.00
val#in geur	.00	.00	.00	.00
val#in zicht	.00	.00	.00	.00

Table 12.1: Scores for MWE candidate *in de smaak vallen* and other nouns in the same cluster

Table 12.1 gives the scores for the MWE *in de smaak vallen*, together with some other nouns that are present in the same cluster. $A_{v\to n}$ shows that there is a clear preference (.12) of the verb *val in* for the noun *smaak*. $R_{v\to n}$ shows that there is a unique preference of the verb for the particular noun *smaak*. For the other nouns (*geur, zicht, ...*), the verb has no preference whatsoever. Therefore, the ratio of verb preference for *smaak* compared to the other nouns in the cluster is 1.00.

$A_{n\to v}$ and $R_{n\to v}$ show similar behaviour. There is a preference (.04) of the noun *smaak* for the verb *val in*, and this preference is unique (1.00).

Table 12.2 gives the scores for the instance *in de put vallen* – which is not a MWE – together with other nouns from the same cluster. The results are quite different from the ones in table 12.1. $A_{v\to n}$ – the preference of the verb for the noun – is quite low in most cases, the highest score being a score of .04 for *gat*. Furthermore, $R_{v\to n}$ does not show a unique preference of *val in* for *put* (a low ratio score of .05). Instead, the preference mass is divided among the various nouns in

MWE candidate	$A_{v\rightarrow n}$	$R_{v\rightarrow n}$	$A_{n\rightarrow v}$	$R_{n\rightarrow v}$
val#in put	.00	.05	.00	.05
val#in kuil	.01	.11	.02	.37
val#in kloof	.00	.02	.00	.03
val#in gat	.04	.71	.01	.24

Table 12.2: Scores for MWE candidate *in de put vallen* and other nouns in the same cluster

the cluster, the highest preference of *val in* being assigned to the noun *gat* (.71).[5]

The other two scores show again a similar tendency; $A_{n\rightarrow v}$ – the preference of the noun for the verb – is low in all cases, and when all nouns in the cluster are considered ($R_{n\rightarrow v}$), there is no 'unique' preference of one noun for the verb *val in*. Instead, the preference mass is divided among all nouns in the cluster.

After assessing the values of the four different measures, our method would propose *in de smaak vallen* as a non-compositional expression and therefore, MWE; on the other hand, the method would consider *in de put vallen* as compositional, thus a non-MWE.

12.4 Results and evaluation

In this section, our automatic method is extensively evaluated. In the first part, we present the results of our quantitative evaluation – including both an automatic evaluation (using Dutch lexical resources) and a manual evaluation (carried out by human judges). The second part is a qualitative evaluation, indicating the advantages and the drawbacks of our method.

12.4.1 Quantitative evaluation

12.4.1.1 Automatic evaluation

The MWEs that are extracted with the fully unsupervised method described above are automatically evaluated by comparing the extracted MWEs to handcrafted lexical databases. Since we have extracted Dutch MWEs, we are using the two Dutch resources available: the Referentie Bestand Nederlands (RBN, (Martin and Maks 2005)) and the Van Dale Lexicographical Information System (VLIS) database. Precision and recall are calculated with regard to the MWEs that are present in our evaluation resources. Among the MWEs in our reference data, we consider only those expressions that are present in our frequency matrix: if the verb is not among the 5,000 most frequent verbs, or the noun is not among the

[5]Note that this expression is ambiguous: it can be used in a literal sense (*in een gat vallen*, 'to fall down a hole') and in a metaphorical sense (*in een zwart gat vallen*, 'to get depressed after a joyful or busy period').

10,000 most frequent nouns, the frequency information is not present in our input data. Consequently, our algorithm would never be able to find those MWEs.

The first six rows of table 12.3 show precision, recall and f-measure for various parameter thresholds with regard to the measures $A_{v\to n}$, $R_{v\to n}$, $A_{n\to v}$ and $R_{n\to v}$, together with the number of candidates found (n). The last line shows the highest values we were able to reach by using the lexical fixedness score.

parameters					precision	recall	f-measure
$A_{v\to n}$	$R_{v\to n}$	$A_{n\to v}$	$R_{n\to v}$	n	(%)	(%)	(%)
.10	.80	–	–	3175	16.09	13.11	14.45
.10	.90	–	–	2655	17.59	11.98	14.25
.10	.80	–	.80	2225	19.19	10.95	13.95
.10	.90	–	.90	1870	20.70	9.93	13.42
.10	.80	.01	.80	1859	20.33	9.69	13.13
.20	.99	.05	.99	404	38.12	3.95	7.16
$Fixedness_{lex}(v,n)$	3.00			3899	15.14	9.92	11.99

Table 12.3: Evaluation results compared to RBN & VLIS

Using only two parameters – $A_{v\to n}$ and $R_{v\to n}$ – gives the highest f-measure ($\pm$ 14%), with a precision and recall of about 17% and about 12% respectively. Adding parameter $R_{n\to v}$ increases precision but degrades recall, and this tendency continues when adding both parameters $A_{n\to v}$ and $R_{n\to v}$. In all cases, a higher threshold increases precision but degrades recall. When using a high threshold for all parameters, the algorithm is able to reach a precision of $\pm$ 38%, but recall is low ($\pm$ 4%).

The lexical fixedness score is able to reach an f-measure of $\pm$ 12% (using a threshold of 3.00). These scores show the best performance that we have reached using lexical fixedness.

12.4.1.2 Human evaluation

The evaluation procedure described above was applied fully automatically by comparing the output of our method to two existing Dutch lexical databases. We are aware of the fact that the automated annotation process may introduce some errors. There may be extracted expressions wrongly labeled as true MWEs but also extracted expressions erroneously labeled as false MWEs. Furthermore, it is known that the used lexical databases are static resources that are likely to miss actual MWEs found in large corpora. This is either because the lexical resources are incomplete, or because the MWEs were not included due to a different understanding of the concept of MWE. With this motivation, we set up a human evaluation experiment. From the dataset that produced the best f-measure ($A_{v\to n}$ = .10 and $R_{v\to n}$ = .80), 200 expressions were semi-randomly selected. To assess the performance

of our method across different frequency ranges, we selected 100 high frequent MWE candidates (frequency ≥ 100) and 100 low frequent ones (frequency < 100).

Three human judges were asked to label the expressions as MWE or as non-MWE. The judges were asked to always provide an answer. To investigate if the rankings from the 3 judges agreed, we employed the Kappa statistic (Cohen 1960). We obtained an average pairwise interannotator agreement of $\kappa = .60$, showing a reasonable correlation between the judges.

The scores assigned by the judges differed severely with regard to frequency range. In the high frequency range, our method was given an average precision of 33.00%. In the low frequency range, precision dropped down to 6.67%. In §12.4.2.2, the results of our human evaluation are evaluated more extensively.

12.4.2 Qualitative evaluation

In this section, we elaborate upon advantages and disadvantages of our semantics-based MWE extraction algorithm by examining the output of the procedure, and looking at the characteristics of the correct MWEs found and the errors made by the algorithm.

12.4.2.1 Advantages of the method

First of all, our algorithm is able to filter out grammatical collocations that cause problems in traditional MWE extraction paradigms. Two examples are given in (5) and (6).

(5) benoemen tot minister, secretaris-generaal
appoint to minister, secretary-general
appoint s.o. {*minister, secretary-general*}

(6) voldoen aan eisen, voorwaarden
meet to demands, conditions
meet the {*demands, conditions*}

In traditional MWE extraction algorithms, based on collocations, highly frequent expressions like the ones in (5) and (6) often get classified as a MWE, even though they are fully compositional. Such algorithms correctly identify a strong lexical affinity between two component words (*voldoen, aan*), which make up a grammatical collocation; however, they fail to capture the fact that the noun may be filled in by a semantic class of nouns. Our algorithm filters out those expressions, because semantic similarity is taken into account.

Our quantitative evaluation shows that the algorithm reaches the best results (i.e. the highest f-measures) when only two parameters ($A_{v \rightarrow n}$ and $R_{v \rightarrow n}$) are taken into account. But upon closer inspection of the output, we have noticed that $A_{n \rightarrow v}$ and $R_{n \rightarrow v}$ are often able to filter out non-MWEs like the expressions b in (7) and (8).

(7) a. op toneel verschijnen
on stage appear
to appear
b. op toneel zingen
on stage sing
to sing on the stage

(8) a. in geheugen liggen
in memory lie
be in memory
b. in ziekenhuis liggen
in hospital lie
lie in the hospital

When only taking into account the first two measures (a unique preference of the verb for the noun), the expressions in b do not get filtered out. It is only when the two other measures (a unique preference of the noun for the verb) are taken into account that they are filtered out – either because the preference of the noun for the verb is very low, or the noun preference for the verb is more evenly distributed among the cluster. The b expressions, which are non-MWEs, result from the combination of a verb with a highly frequent PP. These PPs are typically locative, directional or predicative PPs, that may combine with numerous verbs.

Also, expressions like the ones in (9), where the fixedness of the expression lies not so much in the verb-noun combination, but more in the PP part (*naar school*, *naar huis*) are filtered out by the latter two measures. These preposition-noun combinations seem to be institutionalized PPs, so-called determinerless PPs (Baldwin et al. 2006).

(9) a. naar school willen
to school want
want to go to school
b. naar huis willen
to home want
want to go home

12.4.2.2 Errors of the method

In this section, we give an exhaustive list of the errors made by our algorithm, and quantitatively evaluate the importance of each error category.

1. First of all, our algorithm highly depends on the quality of the noun clustering. If a noun appears in a cluster with unrelated words, the measures will overrate the semantic uniqueness of the expressions in which the noun appears.

2. Syntax might play an important role. Sometimes, there are syntactic restrictions between the preposition and the noun. A noun like *pagina* ‘page’ can only appear with the preposition *op* ‘on’, as in *lees op pagina* ‘read on

page'. Other, semantically related nouns, such as *hoofdstuk* 'chapter', prefer *in* 'in'. Due to these restrictions, the measures will again overrate the semantic uniqueness of the expression.

3. We found many expressions in which the fixedness of the expression lies not so much in the combination of the verb and the prepositional phrase, but rather in the prepositional phrase itself (*naar school*, *naar huis*). Note, however, that our two latter measures were able to filter out many of those expressions (as noted in §12.4.2.1). But in our error evaluation, we used the result that yields the highest f-measure (and does not take the latter measures into account).

4. Our hard clustering method does not take polysemous nouns into account. A noun can only occur in one cluster, ignoring other possible meanings. *Schaal*, for example, means 'dish' as well as 'scale'. In our clustering, it only appears in a cluster of dish-related nouns. Therefore, expressions like *maak gebruik op [grote] schaal* 'make use of [sth.] on a [large] scale', receive again overrated measures of semantic uniqueness, because the 'scale' sense of the noun is compared to nouns related to the 'dish' sense.

5. Related to the previous error category is the fact that certain nouns – although occurring in a perfectly sound cluster – possess a semantic feature or characteristic that distinguishes them from the other nouns in the cluster, and causes the verb to uniquely prefer that particular noun. An example of this kind of error is the expression *eet in restaurant* 'eat in a restaurant', which is perfectly compositional. But due to the fact that the noun *restaurant* ends up in a cluster with nouns such as *bar* 'bar', *café* 'bar', *kroeg* 'pub', *winkel* 'shop', *hotel* 'hotel' – which are places where one is less likely to eat – the fixedness of the expression is overestimated.

6. The effectiveness of our method is highly dependent on the corpus distribution. Sometimes, expressions that would be effective counterweights for the erroneous classification of compositional expressions as MWE just are not found in the corpus. This might be either due to sparseness of the data, or due to the specific nature of the corpus itself. Examples are *sluit wegens verbouwing* 'close due to alteration', with cluster members such as *restauratie* 'restoration' and *renovatie* 'renovation', and *uit van emotie* 'express emotion', with cluster members such as *agressie* 'agression', *irritatie* 'irritation', *ongeduld* 'impatience'. Expressions such as *sluit wegens renovatie* or *uit van irritatie* are perfectly possible, but are not (sufficiently) attested in the corpus. Therefore, the compositional forms which are attested in the corpus are overestimated as MWE.

7. Finally, misclassifications may be caused by parsing errors or other technical issues.

In order to get a better view of the errors of the method, we manually classified the expressions that were evaluated as non-MWE by our judges. Each expression

was assigned to one of the error categories described above. Overall results, and results for high and low frequency expressions are given.

		overall (%)	high freq. (%)	low freq. (%)
1	erroneous clustering	3.6	3.8	3.4
2	specific preposition	6.4	15.4	1.1
3	PP fixedness	26.4	21.2	29.5
4	polysemous word	15.7	13.5	17.0
5	specific semantic feature	22.9	30.8	18.2
6	corpus distribution	21.4	13.5	26.1
7	parsing/other	3.6	1.9	4.5

Table 12.4: Quantitative error evaluation

Misclassifications due to erroneous clustering or parsing errors only constitute a small part of the errors. Also, misclassifications due to syntactic restrictions (specific prepositions) are responsible for only a small part of the errors. More important are misclassifications due to fixedness in the PP, or due to polysemy or specific semantic features of the nouns. The former might be remedied by a more effective use of our measures $A_{n \to v}$ and $R_{n \to v}$, the latter by taking on a soft clustering approach. Finally, there are quite some errors due to the specific distribution of MWEs in the corpus. These errors are more common in the low frequency range. Clearly, our method is highly dependent on the corpus that is used, and it should be sufficiently large in order to adequately classify less frequent MWEs.

12.4.2.3 MWE fuzziness

A last observation to mention is that the status of certain expressions extracted with our method is unclear. Expressions such as *vraag met klem* 'ask with emphasis' or *ga over tot orde [van de dag]* 'pass to the order [of the day]' seem to be on the border of compositionality vs. non-compositionality, and therefore cannot be adequately qualified as MWE or non-MWE. This observation is confirmed by the conflicting views the three judges showed when assessing these kind of expressions.

12.5 Conclusions and further work

Our algorithm based on non-compositionality explores a new approach aimed at large-scale MWE extraction. Using only two parameters, $A_{v \to n}$ and $R_{v \to n}$, yields the highest f-measure. Using the two other parameters, $A_{n \to v}$ and $R_{n \to v}$, increases precision but degrades recall. Due to the formalization of the intuition of non-compositionality (using an automatic noun clustering), our algorithm is able to rule out various expressions that are coined MWEs by traditional algorithms.

Note that our algorithm has taken on a purely semantics-based approach. 'Syntactic fixedness' of the expressions is not taken into account. Combining our semantics-based approach with other MWE extraction methods that take into account different features might improve the results significantly.

We conclude with some issues saved for future work. First of all, we would like to combine our semantics-based method with other methods that are used to find MWEs (especially syntax-based methods), and implement the method in general classification models (decision tree classifier and maximum entropy model). This includes the use of a more principled (machine learning) framework in order to establish the optimal threshold values, and the use of appropriate median values and confidence intervals in order to model the different levels within a continuum of compositionality.

Next, we would like to investigate a number of topics to improve on our semantics-based method. First of all, using the top *k* similar nouns for a certain noun – instead of the cluster in which a noun appears – might be more beneficial to get a grasp of the compositionality of MWE candidates. Also, making use of a verb clustering in addition to the noun clustering might also help in determining the non-compositionality of expressions. Disambiguating among the various senses of nouns should also be a useful improvement. Furthermore, we would like to generalize our method to other syntactic patterns (e.g. verb object combinations), and test the approach for English.

We believe that our method provides a genuine and successful approach to get a grasp of the non-compositionality of MWEs in a fully automated way. We also believe that it is one of the first methods able to extract MWEs based on non-compositionality on a large scale, and that traditional MWE extraction algorithms will benefit from taking this non-compositionality into account.

Acknowledgements

This research was carried out as part of the IRME STEVIN research project. We would like to thank our three human judges (Nicole Grégoire, Jori Mur, Gertjan van Noord) and the two anonymous reviewers for their helpful comments on an earlier version of this paper.

References

Baldwin, T.(2006), Compositionality and Multiword Expressions: Six of One, Half a Dozen of the Other?, Invited talk given at the COLING/ACL'06 Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties.

Baldwin, T., Bannard, C., Tanaka, T. and Widdows, D.(2003), An Empirical Model of Multiword Expressions Decomposability, *Proc. of the ACL-2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan, pp. 89–96.

Baldwin, T., Beavers, J., van der Beek, L., Bond, F., Flickinger, D. and Sag, I.(2006), *In search of a systematic treatment of Determinerless PPs*, Computational Linguistics Dimensions of Syntax and Semantics of Prepositions, Kluwer Academic, pp. 163–180.

Church, K., Gale, W., Hanks, P. and Hindle, D.(1991), Using statistics in lexical analysis, *in* U. Zernik (ed.), *Lexical Acquisition: Exploiting On-line resources to build a lexicon*, Lawrence Erlbaum Associates, New Jersey, pp. 115–164.

Cohen, J.(1960), A coefficient of agreement for nominal scales, *Educational and Psychological Measurement* **20**, 37–46.

Fazly, A. and Stevenson, S.(2006), Automatically constructing a lexicon of verb phrase idiomatic combinations, *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, Trento, Italy.

Katz, G. and Giesbrecht, E.(2006), Automatic identification of non-compositional multi-word expressions using Latent Semantic Analysis, *Proc. of the COLING/ACL'06 Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, Sydney, Australia, pp. 12–19.

Lin, D.(1998), Automatic retrieval and clustering of similar words, *Proceedings of* COLING/ACL *98*, Montreal, Canada.

Lin, D.(1999), Automatic identification of non-compositional phrases, *Proceedings of ACL-99*, University of Maryland, pp. 317–324.

MacQueen, J. B.(1967), Some methods for classification and analysis of multivariate observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, University of California Press, Berkeley, pp. 281–297.

Martin, W. and Maks, I.(2005), *Referentie Bestand Nederlands. Documentatie*.

McCarthy, D., Keller, B. and Carroll, J.(2003), Detecting a Continuum of Compositionality in Phrasal Verbs, *Proc. of the ACL-2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan.

Ordelman, R.(2002), Twente Nieuws Corpus (TwNC). Parlevink Language Techonology Group. University of Twente.

Pearce, D.(2001), Synonymy in collocation extraction, *WordNet and Other lexical resources: applications, extensions & customizations (NAACL 2001)*, Carnegie Mellon University, Pittsburgh, pp. 41–46.

Piao, S., Rayson, P., Mudraya, O., Wilson, A. and Garside, R.(2006), Measuring MWE compositionality using semantic annotation, *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, Association for Computational Linguistics, Sydney, Australia, pp. 2–11.

Resnik, P.(1993), *Selection and Information: A Class-Based Approach to Lexical Relationships*, PhD Thesis, University of Pennsylvania.

Resnik, P.(1996), Selectional constraints: An information-theoretic model and its computational realization, *Cognition* **61**, 127–159.

Sag, I., Baldwin, T., Bond, F., Copestake, A. and Flickinger, D.(2002), Multiword

Expressions: a pain in the neck for NLP, *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico, pp. 1–15.

van der Plas, L. and Bouma, G.(2005), Syntactic contexts for finding semantically similar words, *Computational Linguistics in the Netherlands 2004. Selected Papers from the Fifteenth CLIN Meeting* pp. 173–184.

van Noord, G.(2006), At Last Parsing Is Now Operational, *in* P. Mertens, C. Fairon, A. Dister and P. Watrin (eds), *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, Leuven, pp. 20–42.

Venkatapathy, S. and Joshi, A.(2005), Measuring the relative compositionality of verb-noun collocations by integrating features, *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, pp. 899–906.

Villada Moirón, B. and Tiedemann, J.(2006), Identifying idiomatic expressions using automatic word-alignment, *Proceedings of the EACL 2006 Workshop on Multi-word-expressions in a multilingual context"*, Trento, Italy, pp. 33–40.

Weeds, J.(2003), *Measures and Applications of Lexical Distributional Similarity*, PhD Thesis, University of Sussex.

Wu, Z. and Palmer, M.(1994), Verb semantics and lexical selection, *32nd. Annual Meeting of the Association for Computational Linguistics*, New Mexico State University, Las Cruces, New Mexico, pp. 133–138.

13

An efficient memory-based morphosyntactic tagger and parser for Dutch

Antal van den Bosch[†]*, Bertjan Busser*[†]*, Sander Canisius*[†]*, and Walter Daelemans*[‡]
[†]Tilburg University
[‡]University of Antwerp

Abstract

We describe TADPOLE, a modular memory-based morphosyntactic tagger and dependency parser for Dutch. Though primarily aimed at being accurate, the design of the system is also driven by optimizing speed and memory usage, using a trie-based approximation of k-nearest neighbor classification as the basis of each module. We perform an evaluation of its three main modules: a part-of-speech tagger, a morphological analyzer, and a dependency parser, trained on manually annotated material available for Dutch – the parser is additionally trained on automatically parsed data. A global analysis of the system shows that it is able to process text in linear time close to an estimated 2,500 words per second, while maintaining sufficient accuracy.

13.1 Introduction

In this paper we introduce TADPOLE (TAgger, Dependency Parser, and mOrphoLogical analyzEr), a modular morpho-syntactic tagger, analyzer and parser

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

for Dutch. In designing TADPOLE we aim for three partially competing goals: (1) high accuracy, (2) high and preferably linear processing speed, and (3) low memory usage. TADPOLE is particularly targeted at the increasing need for fast, automatic NLP systems applicable to very large (multi-million to billion word) document collections that are becoming available due to the progressive digitization of both new and old textual data. This scale does not fit well with systems that perform exponentially in terms of the length of their input, spending perhaps minutes on single sentences, and neither with linear-time but slow processing system that would take, e.g., a second per word – which would imply more than ten days to process just one million words of text.

Rather than a mix of methods, we opt for a single processing engine to be used in all modules to simplify the software engineering aspects. As the core engine we chose memory-based learning, in particular a fast trie-based approximation of k-nearest neighbor classification, IGTREE (Daelemans et al. 1997a). Memory-based learning has been shown to produce competitive, state-of-the-art performance in part-of-speech tagging (Daelemans et al. 1996) and morphological analysis (Van den Bosch and Daelemans 1999), and has recently also been employed in a dependency parser (Canisius et al. 2006) with some initial success. IGTREE has been shown to speed up normal k-nearest neighbor classification several orders of magnitude, while retaining much of its generalization accuracy. With IGTREE we aim to reach high processing speed (an aspect of goal 2) and low memory usage (goal 3); the accuracy levels (goal 1) are expected to be lower than those of k-nearest neighbor classification; empirical tests are needed to ascertain the gap.

Linear processing speed (another aspect of goal 2) is straightforwardly achieved with memory-based part-of-speech tagging and morphological analysis; both approaches are fully linear in their default sequence processing method (Daelemans et al. 1996, Van den Bosch and Daelemans 1999). With dependency parsing, however, linearity is an issue. The approach proposed by Canisius et al. (2006) involves a processing step that is quadratic in principle, but linearly bounded, and a deterministic search through the predicted dependency relations.

In this paper we first lay out the architecture of the system in Section 13.2. We then provide evaluations of the three modules in Section 13.3, and we evaluate the system globally in Section 13.4. Related work is discussed in Section 13.5. We close the paper with a discussion of future work in Section 13.6.

13.2 Architecture

The intended function of TADPOLE is to automatically annotate Dutch text with morpho-syntactic information at the word level, and syntactic dependency relations between words at the sentence level. To enable a proper treatment of incoming text, a tokenizer is used for preprocessing. We adopted a rule-based tokenizer that splits punctuation markers from words, using seed lists of common Dutch abbreviations, and that splits sentences according to a set of heuristic rules (Reynaert 2007). Tokenized text is then fed to the part-of-speech tagger and the morphological analyzer. Subsequently, predicted part-of-speech tags are for-

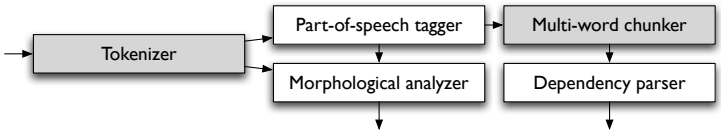


Figure 13.1: Schematic architecture of TADPOLE. The grey boxes represent non-machine-learning-based modules.

warded to the morphological analyzer, which uses the tags to choose among the analyses it has generated for ambiguous words. The tags are also used as input to the dependency parser, which in turn demands that a fixed list of multi-word phrases and all multi-word proper nouns are collated by a straightforward lookup-based multi-word chunker.

Figure 13.1 schematically illustrates the information flow of the processing modules. Each memory-based module (the white boxes) uses a classification engine that converts its input to a partial output; each conversion step is one classification of a windowed snapshot of the input sequence into an output label. Sequences of output labels are gathered until the end of the word or sentence, and subsequently converted into a full output (per word for the morphological analyzer, and per sentence for the part-of-speech tagger and dependency parser). Section 13.3 provides more detailed information on the functioning of each module.

The classifier engine in the three memory-based processing modules is IGTREE (Daelemans et al. 1997a), an algorithm for the top-down induction of decision trees. It compresses a database of labeled examples into a lossless-compression decision-tree structure that preserves the labeling information of all examples, and technically should be named a *trie* according to Knuth (1973). A labeled example is a feature-value vector encoding input (in our case, windowed subsequences of letters, words, or part-of-speech tags) and output (in our case, labels encoding morphological information, part-of-speech tags, or syntactic dependency relation types).

An IGTREE is a hierarchical tree composed of nodes that each represent a partition of the original example database, and are labeled by the most frequent class of that partition. Besides a majority class label, the nodes also hold complete counts of all class labels in the database partition they represent. The root node of the trie thus (1) represents the entire example database, (2) carries the most frequent value as class label, and (3) holds the occurrence counts of all classes in the full training set. In contrast, end nodes (leafs) represent a homogeneous partition of the database in which all examples have the same class label; the node merely stores this label along with the size of the homogeneous partition. Non-ending nodes branch out to nodes at deeper levels of the trie. Each branch represents a test on a feature value; branches fanning out of one node test on values of the same feature.

To attain high compression levels, IGTREE branches out from the root node by testing on the most informative, or most class-discriminative feature first, followed at the next level by the second-most discriminative feature. IGTREE uses information gain (IG) to estimate discriminativeness. The IG of feature i is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature with respect to predicting the class label: $IG_i = H(C) - \sum_{v \in V_i} P(v) \times H(C|v)$, where C is the set of class labels, V_i is the set of values for feature i, and $H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$ is the entropy of the class labels. IGTREE computes the IG of all features once on the full database of training examples, makes a feature ordering once on these computed IG values, and uses this ordering throughout the whole trie.

IGTREE effectively performs a lossless compression of the labeling information of the original example database. As long as the database does not contain fully ambiguous examples (with the same features, but different class labels), the trie produced by IGTREE is able to reproduce the classifications of all examples in the original example database perfectly.

13.3 Modules

We describe for each of the three IGTREE-based modules how their tasks are encoded into classification tasks, and provide estimates of their generalization performance on unseen words and text.

13.3.1 Part-of-speech tagging

The approach to part-of-speech tagging taken in TADPOLE was originally introduced by Daelemans et al. (1996). The proposed tagger is a combination of a submodule that disambiguates the tags of words it has seen before, given their context, and a submodule that predicts tags to words it has not seen before. Both taggers process from left to right, and use windowing to represent the local context around the word to be tagged. The left part of the window also includes the joint tagger's previously predicted tags, while in the right part of the window the yet ambiguous tags of the known right neighboring words are incorporated.

The second submodule, the *unknown words* tagger, cannot use the word in focus as a predictive feature since it has not seen it before, but some surface features of the word are represented. Furthermore, both taggers are helped by converting low-frequency words to more generic placeholder strings that retain some of their surface features. Also, the unknown words tagger is not trained on the full training set, but rather on a subset of low-frequency words in their context in the training set, as they are the most representative of actual unseen words, which will tend to occur in the same frequency band. In detail, the features for the two subtaggers are the following:

- For the *known words* tagger: the focus word and its immediate left and right neigboring words, the three preceding predicted tags, and the two still ambiguous tags to the right.

Task	Full tag	Main tag
Known words	96.8	98.7
Unknown words	76.4	84.3
All words	96.5	98.6

Table 13.1: Percentages of correctly tagged test words, overall (bottom line) and split into known words and unknown words, on the full tag and on the main tag only.

- For the *unknown words* tagger: the first two letters and the last three letters of the focus word; binary features marking whether the word is capitalized, contains a hyphen, or one or more numbers; its immediate left and right neighboring words; the three preceding predicted tags, and the two still ambiguous tags at the right.

When trained on a substantial training corpus, often less than 10% (or even less than 5%) of words in new text will not have occurred in the training corpus. Hence, the first submodule, the *known words* tagger, is responsible for a major part of the work. Yet, the remaining work for the unknown word tagger is harder. For the TADPOLE part-of-speech tagger we opted to use IGTREE for the known words tagger, but use TRIBL for the unknown words tagger. TRIBL is a hybrid between the fast approximation IGTREE and the slower IB1-IG algorithm that implements k-nearest neighbor in its unabridged form (Daelemans et al. 1997b)[1]; it builds a trie structure for the most informative features, and performs k-nearest neighbor classification on the remaining features. For building the tagger, the Mbt wrapper was used[2].

The data used for training the TADPOLE tagger consists of a broad selection of available manually annotated part-of-speech tagged corpora for Dutch tagged with the Spoken Dutch Corpus tagset (Van Eynde 2004): The approximately nine-million word of the transcribed Spoken Dutch Corpus itself (Oostdijk et al. 2002), the ILK corpus with approximately 46 thousand part-of-speech tagged words, the D-Coi corpus with approximately 330 thousand words, and the 754-thousand word Eindhoven corpus (Uit den Boogaart 1975) which has been automatically retagged with the Spoken Dutch Corpus tagset. Together this accounts for 10,979,827 manually-checked part-of-speech tagged words, all using the same rich tagset of 316 tags.

We split this 10 million-word corpus randomly (at the sentence level) into a 90% training set and a 10% test set. The performance of the tagger on known words and unknown words in the test set, as well as on all test words, is listed in Table 13.1. Not surprisingly, the tagger has significantly more trouble tagging unknown words. The Spoken Dutch Corpus tagset makes a distinction between the

[1] IGTREE, TRIBL, and IB1-IG are included in the TiMBL software package, version 5.1, available from `http://ilk.uvt.nl/timbl`.
[2] Mbt, version 2.0.1: `http://ilk.uvt.nl/mbt`.

main tag (a traditional 12-tag distinction) and the morphosyntactic subtags, which are not always used in higher-level applications; the generalization accuracy on the main tag reaches a respectable 98.6%.

In the overall tagging accuracy, the influence of the unknown-word tagger is of course related to the amount of unknown words in the text to be tagged. In the 10% test set, about 98.8% of all tokens is also present in the 90% training set, but this test is a sentence-level partition of the same texts as the training set is drawn from. Typically, coverage of tokens in a randomly selected text from outside the (genres of the) training set will be somewhat lower, as illustrated by the following two examples. A first random text, offering general instructions on Unix, containing many foreign words and command line fragments, is covered by 89.8%. The second text, the full text of the novel *Het boetekleed*, a Dutch translation of Ian McEwen's *Atonement*, is covered by 97.9%.

13.3.2 Morphological analysis

We take the task of analyzing the morphology of Dutch words to include (1) segmenting a wordform into its morphemes; (2) labeling each morpheme with its function (e.g. a stem with a certain part-of-speech tag, or being a derivational affix, or an inflection), and (3) identifying all spelling changes between the wordform and its underlying morphemes (Van den Bosch and Daelemans 1999). We draw our examples from the CELEX lexical database (Baayen et al. 1993), which features a full morphological analysis for 363,690 of them. We took each wordform and its associated analysis, and created task examples using a windowing approach, which transforms each wordform into as many examples as it has letters. Each example focuses on one letter, and includes a fixed number of left and right neighbor letters, chosen here to be five. Consequently, each example spans eleven letters, which is also the average word length in the CELEX database.

To illustrate the construction of examples, Table 13.2 displays the 15 examples derived from the Dutch example word abnormaliteiten (abnormalities) and their associated classes. The class of the first example is "A", which means that the morpheme starting in a is an adjective ("A"). This morpheme continues up to the eighth example, which is labeled with "0+Da", meaning that at that position, an a is deleted from the underlying morpheme. The coding thus tells that the first morpheme is the adjective abnormaal. The second morpheme, iteit, has class "N_A$*$". This complex tag indicates that when iteit attaches right to an adjective (encoded by "A$*$"), the new combination becomes a noun ("N_"). Finally, the third morpheme is en, which is a plural inflection (labeled "m" in CELEX).

This way we generated a database of 3,209,064 examples. Within these examples, 3,806 different class labels occur. The most frequently occurring class label is "0", occurring in 69.3% of all instances. The three most frequent non-null labels are "N" (6.9%), "V" (4.2%), and "A" (1.3%).

When a wordform is listed in CELEX as having more than one possible morphological labeling (e.g., a morpheme may be N or V, the inflection -en may be plural for nouns or infinitive for verbs), these labels are joined into ambiguous

instance number	left context					focus letter	right context					TASK
1	_	_	_	_	_	a	b	n	o	r	m	A
2	_	_	_	_	a	b	n	o	r	m	a	0
3	_	_	_	a	b	n	o	r	m	a	l	0
4	_	_	a	b	n	o	r	m	a	l	i	0
5	_	a	b	n	o	r	m	a	l	i	t	0
6	a	b	n	o	r	m	a	l	i	t	e	0
7	b	n	o	r	m	a	l	i	t	e	i	0
8	n	o	r	m	a	l	i	t	e	i	t	0+Da
9	o	r	m	a	l	i	t	e	i	t	e	N_A*
10	r	m	a	l	i	t	e	i	t	e	n	0
11	m	a	l	i	t	e	i	t	e	n	_	0
12	a	l	i	t	e	i	t	e	n	_	_	0
13	l	i	t	e	i	t	e	n	_	_	_	0
14	i	t	e	i	t	e	n	_	_	_	_	m
15	t	e	i	t	e	n	_	_	_	_	_	0

Table 13.2: Instances with morphological analysis classifications derived from abnormaliteiten, analyzed as $[\text{abnormaal}]_A[\text{iteit}]_{N\_A*}[\text{en}]_m$.

classes (“N/V”). Ambiguity in syntactic and inflectional tags occurs in 3.6% of all morphemes in our CELEX data. When the morphological analyzer generates more than one analysis based on these ambiguous classes, it asks for the part-of-tagger to break the tie – hence the arrow from the tagger to the analyzer in Figure 13.1. We created a translation table between combinations of CELEX main tags and inflectional markers such as “m” on the one hand, and the CGN tags of the part-of-speech tagger on the other hand, to allow matching the CGN tags to the ambiguous analyses. We observed that when the tagger is correct and the analyzer generates the appropriate analyses, the CGN tags predicted by the tagger, with their main tag and the morpho-syntactic subtags, always provide sufficient matches to disambiguate between ambiguous analyses. If due to an error of either module no match is possible to break the tie, a random choice is made.

To evaluate the morphological analyzer, we split the CELEX database randomly in a 90% training set (of 362,690 words, or 2,888,197 examples) and a 10% test set (of 36,369 words, or 320,867 examples). When trained on the full 90% training set, IGTREE correctly segments 79.0% of test words; e.g., it would segment abnormaliteiten correctly into [abnormal][iteit][en]. Also taking into account spelling changes and morpheme types (stems with part-of-speech, affixes, inflections, e.g. $[\text{abnormaal}]_A[\text{iteit}]_{N\_A*}[\text{en}]_m$), 56.3% of all test words are fully correctly analyzed. These generalization accuracies, obtained on a random 10% of CELEX words, can be seen as approximations of the analyzer’s performance on unknown words in free text. Performing a coverage check similar to the one in the previous section, we observe that CELEX covers about 98.3% of the tokens in the test material of the tagger, 83.9% of the Unix instruction document, and 98.1% of

the word tokens in *Het boetekleed*. As IGTREE performs a lossless compression of the training set, the analysis or alternate analyses of any word that is also in CELEX will be flawlessly retrieved; hence, the effective accuracy of the analyzer on a text such as the novel is at least 98.1%, and possibly around 99%, as we estimated that about 56.3% of unknown words receives a correct analysis.

13.3.3 Dependency parsing

In the TADPOLE approach to dependency parsing, IGTREE is trained to predict (directed) labeled dependency relations between a head and a dependent. For each token in a sentence, examples are generated where this token is a potential dependent of each of the other tokens in the sentence. To prevent explosion of the number of classification cases to be considered for a sentence, we restrict the maximum distance between a token and its potential head. We selected this distance so that 95% of the dependency relations in the training data are covered, which is at a maximum distance of eight words. The label that is predicted for each classification case serves two different purposes at once: 1) it signals whether the token is a dependent of the designated head token, and 2) if the instance does in fact correspond to a dependency relation in the resulting parse of the input sentence, it specifies the type of this relation as well.

The features we used for encoding instances for this classification task correspond to a rather simple description of the head-dependent pair to be classified. For both the potential head and dependent, there are features encoding a 1-1-1 window of words and part-of-speech tags predicted by our tagger; in addition, there are two spatial features: a relative position feature, encoding whether the dependent is located to the left or to the right of its potential head, and a distance feature that expresses the number of tokens between the dependent and its head.

Thus, dependency parsing is first broken down into classifications at the level of word-to-word dependency relations. In a second step these relations need to be gathered per sentence to form a dependency tree. A dependency tree is regarded as a set of dependency relations connecting a head and a dependent. For a set of such relations to form a valid dependency tree, some constraints should be satisfied: 1) each token can only be linked as a dependent to maximally one head token (though a token may be a head to more than one dependent), and 2) dependency relations should not form a cycle. As long as these two constraints are satisfied, a dependency tree can be treated as a set of dependency relations without losing any information.

Naively applying this approach results in a number of practical issues however, which may also negatively affect the performance. First, the classification task as formulated gives rise to a highly skewed class distribution in which examples that correspond to a dependency relation are largely outnumbered by "negative" examples. Second, there is a quadratic increase of instances to be classified as sentence length increases, that is, a sentence of n tokens translates to $n(n-1)$ classification cases.

One issue that may arise when considering each potential dependency relation

as a separate classification case is that inconsistent trees are produced. For example, a token may be predicted to be a dependent of more than one head. To recover a valid dependency tree from the separate dependency predictions, a simple inference procedure is performed. Consider a token for which the dependency relation is to be predicted. For this token, a number of classification cases have been processed, each of them indicating whether and if so how the token is related to one of the other tokens in the sentence. Some of these predictions may be negative, i.e. the token is not a dependent of a certain other token in the sentence, others may be positive, suggesting the token is a dependent of some other token.

If all classifications are negative, the token is assumed to have no head, and consequently no dependency relation is added to the tree for this token. If one of the classifications is non-negative, suggesting a dependency relation between this token as a dependent and some other token as a head, this dependency relation is added to the tree. Finally, there is the case in which more than one prediction is non-negative. By definition, at most one of these predictions can be correct; therefore, only one dependency relation should be added to the tree. To select the most-likely candidate from the predicted dependency relations, the candidates are ranked according to the classification confidence of the base classifier that predicted them, and the highest-ranked candidate is selected for insertion into the tree. For example, if in the sentence Ik hoor haar zingen, *I hear her singing*, the word haar is classified as relating to hoor in the "OBJ1" relation (direct object) with confidence 8, and to zingen in the "DET" relation (determiner) with confidence 5, the first prediction is selected, and the second discarded.

As a measure of confidence for the predictions made by IGTREE we divide the tree-node counts assigned to the majority class by the total counts assigned to all classes. Though this confidence measure is rather crude, and should not be confused with any kind of probability, it tends to work quite well in practice (Canisius et al. 2006).

The base classifier in our parser is faced with a classification task with a highly skewed class distribution, i.e. instances that correspond to a dependency relation are largely outnumbered by those that do not. In practice, such a huge number of negative instances usually results in classifiers that tend to predict fairly conservatively, resulting in high precision, but low recall. In the approach introduced above, however, it is better to have high recall, even at the cost of precision. A missed relation by the base classifier can never be recovered by the inference procedure. Also, due to the constraint that each token can only be a dependent of one head, excessive prediction of dependency relations can still be corrected by the inference procedure. An effective method for increasing the recall of a classifier is downsampling of the training data. In downsampling, instances belonging to the majority class (in this case the negative class) are removed from the training data, so as to obtain a more balanced distribution of negative and non-negative instances.

Canisius et al. (2006) describe the effect of systematically removing an increasingly larger part of the negative instances from the training data. They report that downsampling helps to improve recall, at the cost of precision, but indeed im-

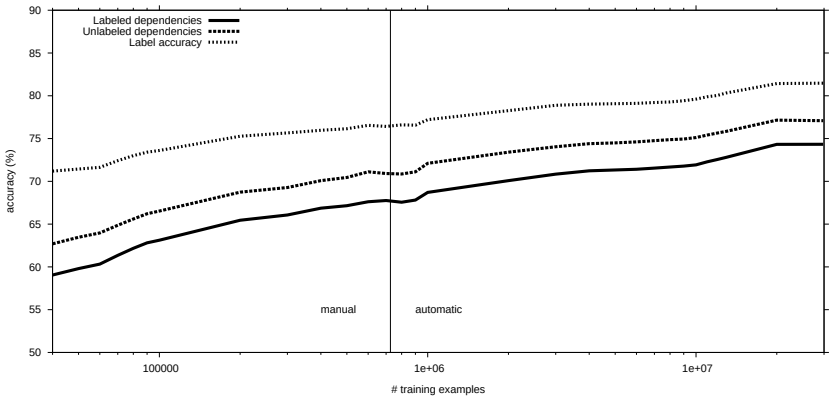


Figure 13.2: Dependency parsing learning curves in terms of correctly labeled dependencies, unlabeled dependencies, and label accuracy.

proving the dependency parser, with a maximal performance at downsampling rate $1:2$ (i.e. twice as many negative examples as positive ones). Note that downsampling is naturally restricted to the training data; the test data is not downsampled as the labeling is not known yet.

As training material for our parser we used all manually annotated data available in the Alpino Treebank[3] (Van der Beek et al. 2001), amounting to 262,452 words, converted to 2,959,456 pairwise examples, and subsequently downsampled to 726,440 examples. We also collected data that is automatically parsed by the Alpino parser (Malouf and Van Noord 2004), available in significantly larger quantities than manually annotated data. We added several millions words of automatically parsed text from Wikipedia pages, newspaper articles, and the full Eindhoven corpus except a portion taken out as test set (see below). We converted this Alpino output to the column format used in the CoNLL-X Shared Task (Buchholz and Marsi 2006), replacing the part-of-speech information generated by Alpino by the output of TADPOLE's tagger described in Subsection 13.3.1. Also in this process, in special cases (particularly with multi-word units and coordinations without a conjunction) multiple heads in the original treebank are discarded, keeping only the leftmost head.

Figure 13.2 displays the learning curves of three commonly used evaluation metrics (Buchholz and Marsi 2006), viz. labeled and unlabeled dependency relation accuracy, and the accuracy on the label per word. The test set consists of 2,530 sentences (47,471 words) taken from the manually parsed section of the Eindhoven corpus (the *cdbl* part); this is newspaper text with relatively long sentences with many subclauses and quotations. The vertical line at 726,400 downsampled pairwise examples marks the transition of manually labeled material to automat-

[3]Alpino Treebank: `http://www.let.rug.nl/ vannoord/trees/`.

	% Correct assignment	
Aspect	Only manual data	Automatic data added
Labeled dependencies	67.3	74.3
Unlabeled dependencies	70.6	77.1
Label accuracy	76.3	81.5

Table 13.3: Percentages of correctly assigned dependencies, with and without labeling, and the accuracy on labels only, trained on the maximal amount of training data, tested on newspaper texts, before and after the addition of automatically parsed training data.

ically parsed data. Despite a dip in performance in all three evaluation metrics, the curves suprisingly return to their trajectories, and continue to rise – albeit at a sub-loglinear rate with increasing amounts of training data. The exact scores of the parser, trained on a current maximum of 29,778,197 examples, and tested on the aforementioned manually parsed test set, are displayed in Table 13.3. At best, the parser identifies and labels dependency relations between words at an accuracy of 74.3.

13.4 Speed and memory usage analysis

Thus far we have not reported on speeds and memory usage, except in passing when comparing the morphological analyzer to IB1-IG. Three design goals of TADPOLE relate to speed and memory: we want the system to be fast, as linear as possible in the length of the input, and costing as little memory as possible. We measured the speed of our classifiers in terms of the number of words they processed per second, and the bytesize of the IGTREES[4]. Table 13.4 summarizes the measurements taken at the maximal sizes of the training sets used in the previous section to estimate the generalization accuracies of each module. The table also lists the speed of the rule-based tokenizer and multi-word chunker for completeness, as these modules do cost some memory[5] and time. As can be seen in the table, the parser consumes most memory, being trained also on the largest amount of training examples (nearly 30 million). The part-of-speech tagger consumes a fair bit of memory as well, due to the TRIBL-based *unknown words* tagger.

Disregarding the fast rule-based preprocessing modules, the tagger is the fastest module with about 10,160 words per second, while the morphological analyzer is the slowest, processing about 6,715 words per second. Given a single processor, the aggregated speed with which TADPOLE can process text with all three modules is about 2,488 words per second. This number assumes single-CPU, full streaming performance.

One remaining design goal is to include a parser with preferably linear performance. We measured the speed and accuracy of the parser on different sentence

[4]The hardware used for testing is equipped with Dual Core AMD Opteron 880 2,412 Mhz processors.
[5]They are implemented as Perl scripts and require the Perl executable at runtime.

Module	Memory (Mb)	1000 words/s
Part-of-speech tagging	23.3	10.1
Morphological analyzer	2.9	6.7
Dependency parser	68.9	7.6
Tokenizer (rule-based)	Perl	81.9
MWU chunker (rule-based)	Perl	120.3
Total	95.1 + Perl	2.5

Table 13.4: Amount of memory used, and numbers of words processed by the five modules at maximal training set sizes. Bottom line sums the amount of memory, and aggregates the speeds.

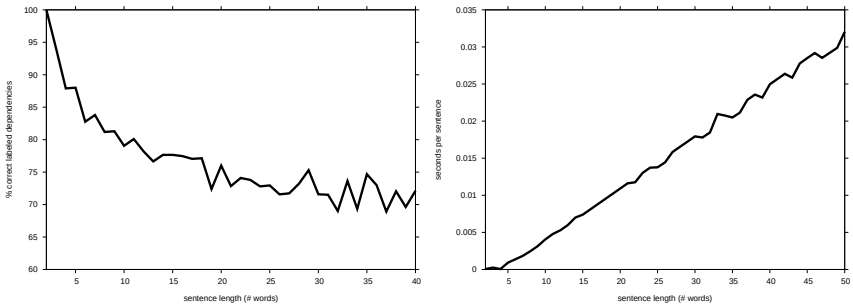


Figure 13.3: Generalization accuracies (left) and seconds per sentence (right) of the dependency parser trained on maximal amounts of data, measured per sentence length from 2 to 50.

lengths found in our test set. Figure 13.3 shows both, measured separately for all sentence lengths from 2 to 50. As the left graph of Figure 13.3 shows, sentences shorter than length 20 are parsed at above-average performance levels. The right graph of Figure 13.3 shows a perhaps more unexpected linear relation between the length of a sentence and the average time it takes to parse it. Earlier we noted that for each sentence pairwise examples are generated ($n(n-1)$, to be exact), but we also constrained this (also with test sentences) to pairs of words within a range of eight words from each other, as 95% of all relations in the training corpus occur within that range. This fixed constraint bounds the number of examples per sentence, making the relation between the sentence length and the number of examples effectively linear.

13.5 Related research

Most if not all related work on morpho-syntactic analysis, tagging, and parsing on Dutch has focused on these tasks in isolation. Schone and Jurafsky (2000) describe

an unsupervised approach to computational morphological analysis, using CELEX as a gold standard. Their knowledge-free method analyzes words in a large corpus above a frequency threshold of 10. Matching these analyses to the ones in CELEX, they report F-scores on correctly identified morphemes of around 79.6. Without a direct comparison, we can safely say that our supervised system vastly outperforms this system, even if we would only look up analyses from CELEX (which their system is obviously not allowed to).

Van Halteren et al. (2001) provide generalization accuracies of various tagging systems trained on Dutch data annotated using the Wotan tagset, a predecessor of, and comparable to, the CGN tagset. Using additional learning methods (hidden markov models, transformation-based learning, and maximum-entropy tagging) and combinations of these taggers in ensemble architectures, but using only the 754-thousand-words Eindhoven corpus, the best cross-validated accuracy reported is 93.3%, and 96.4% using a reduced version of the tagset, "WotanLite"; this is the performance of a stacked ensemble of classifiers. In contrast, with about 10 million words of training data we attain about the same accuracy (96.5%) in a similar experiment with a tagset that is at least as rich as Wotan, but using a single classifier.

Buchholz and Marsi (2006) provide an overview of systems who competed in the CoNLL-X Shared Task, which also used a part of the manually annotated Alpino treebank, split in training data (195,069 words, 13,349 sentences) and test data (5,463 words, 386 sentences). For the best system (McDonald et al. 2006) a labeled dependency score of 79.2 is reported, clearly superior to our 74.3 (obtained with more training data, tested on a different test set). Yet, this best performing system is a more complicated two-stage discriminative parser that first performs unlabeled parsing, and then assigns labels, and runs in cubic time as opposed to our linear parser.

An obvious competitor to our parser is the original Alpino parser (Malouf and Van Noord 2004) which it hopes to emulate. Probably the best parser for Dutch, Alpino is a typical modern example of a rule-based approach that has hybridized with a stochastic, data-driven approach. After a rule-based core generates possible parses for a given sentence (possibly hundreds or thousands), a stochastic component searches in this space of possibilities for the most likely parse, where the statistics are derived from the Alpino treebank.

Alpino has been evaluated with various metrics; Malouf and Van Noord (2004) argue for using an adapted form of *concept accuracy* to estimate the correctness of the dependency labeling. The labeled dependencies accuracy metric of the CoNLL-X shared task (Buchholz and Marsi 2006), used in this paper, has the same aim; both metrics essentially compute $\#correct/\#total$, i.e., the number of correctly assigned relations divided by the total number of relations. The difference between the two metrics is that Alpino generates a limited amount of non-terminal nodes in its trees, which necessitates their metric, where in our case the number of generated relations will never be larger than the number of tokens, hence the simple labeled dependency accuracy metric suffices. Given this, we cannot currently compare our parsers to Alpino. Still, it is interesting to contrast some results

obtained on the same or similar test sets. On a similar test set to ours, composed of news articles, Alpino is reported to attain a concept accuracy of 87.9%, which is markedly higher than our 74.3% accuracy on labeled dependencies. On a small corpus of quesions, Alpino attains a concept accuracy of 88.7%; a test of our parser on this corpus yields a labeled dependency accuracy of 78.7%. Clearly, our parser lags behind Alpino in terms of accuracy.

13.6 Discussion

We have described the TADPOLE system, a robust modular morphological analyzer, part-of-speech tagger, and dependency parser for Dutch. Including the classification engine, the complete system costs about 95 Mb of memory, and has an estimated processing speed of close to 2,500 words per second, assuming a common processor type and full streaming performance. The tagger is estimated to be about 96.5% correct on unseen text (98.6% in terms of main tags). The morphological analyzer can segment about 79.0% of unseen words correctly, and can produce a completely correct analysis with part-of-speech tags and spelling changes for 56.3% of unseen words. The coverage of the tagger and the morphological analyzer is quite high; a random novel text is covered at about 98% of all tokens. In the case of the morphological analyzer this means that it is able to losslessly reproduce correct analyses for at least these 98% tokens. The dependency parser, feeding on tags generated by the part-of-speech tagger, generates dependency relations between pairs of words at an accuracy rate of about 74.3%. The parser is observed to parse in linear time in function of the length of the input; although it has a quadratic component in the example generation process, this process is constrained by a threshold that makes the number of examples linear in the length of the sentence.

In future work we aim to prolong the learning curve of the dependency parser, as much more training data is still available. If the learning curve does not flatten too much it may be possible in the long run to develop a linear-time memory-based emulation of the Alpino parser. We may introduce some extra internal flow of information, such as from the morphological analyzer to the unknown-words module of the part-of-speech tagger. Other future work involves the incorporation of other modules into TADPOLE such as a named-entity recognizer, a semantic role labeler, and a co-reference module, so that the abbreviation will stand for TAgger, Dependency Parser, and Other Language Engines.

Acknowledgements

We gratefully acknowledge the contributions to the different modules of Sabine Buchholz (the tokenizer), Jakub Zavrel (the tagger), Ko van der Sloot (the Timbl software), and Ton Weijters (the IGTREE algorithm). Thanks also to the anonymous reviewers for valuable suggestions. We are indebted to Gertjan van Noord and his co-workers for their invaluable work on the Alpino parser. This work was funded by NWO, The Netherlands Organisation for Scientific Research, as part of

the IMIX Program and the Vici project “Implicit Linguistics”.

References

Baayen, R. H., Piepenbrock, R. and van Rijn, H.(1993), *The CELEX lexical data base on CD-ROM*, Linguistic Data Consortium, Philadelphia, PA.

Buchholz, S. and Marsi, E.(2006), CoNLL-X shared task on multilingual dependency parsing, *Proceedings of CoNLL-X, the Tenth Conference on Computational Natural Language Learning*, New York, NY.

Canisius, S., Bogers, T., Van den Bosch, A., Geertzen, J. and Tjong Kim Sang, E.(2006), Dependency parsing by inference over high-recall dependency predictions, *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL-X*, New York, NY.

Daelemans, W., Van den Bosch, A. and Weijters, A.(1997a), IGTree: using trees for compression and classification in lazy learning algorithms, *Artificial Intelligence Review* **11**, 407–423.

Daelemans, W., Van den Bosch, A. and Zavrel, J.(1997b), A feature-relevance heuristic for indexing and compressing large case bases, *in* M. Van Someren and G. Widmer (eds), *Poster Papers of the Ninth European Conference on Machine Learing*, University of Economics, Prague, Czech Republic, pp. 29–38.

Daelemans, W., Zavrel, J., Berck, P. and Gillis, S.(1996), MBT: A memory-based part of speech tagger generator, *in* E. Ejerhed and I. Dagan (eds), *Proceedings of the Fourth Workshop on Very Large Corpora*, ACL SIGDAT, pp. 14–27.

Knuth, D. E.(1973), *The art of computer programming*, Vol. 3: Sorting and searching, Addison-Wesley, Reading, MA.

Malouf, R. and Van Noord, G.(2004), Wide coverage parsing with stochastic attribute value grammars, *Proceedings of the IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses*.

McDonald, R., Lerman, K. and Pereira, F.(2006), Multilingual dependency analysis with a two-stage discriminative parser, *in* L. Màrquez and D. Klein (eds), *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL-X*, New York, NY, USA.

Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J., Moortgat, M. and Baayen, H.(2002), Experiences from the spoken dutch corpus project, *in* M. González Rodriguez and C. Paz Suárez Araujo (eds), *Proceedings of the third International Conference on Language Resources and Evaluation*, pp. 340–347.

Reynaert, M.(2007), Sentence-splitting and tokenization in D-Coi, *Technical Report ILK 07-03*, ILK Research Group.

Schone, P. and Jurafsky, D.(2000), Knowledge-free induction of inflectional morphologies, *Proceedings of the North American Chapter of the Association*

of Computational Linguistics, Pittsburgh, PA, USA.

Uit den Boogaart, P.(1975), *Woordfrequenties in geschreven en gesproken Nederlands*, Oosthoek, Scheltema & Holkema, Utrecht.

Van den Bosch, A. and Daelemans, W.(1999), Memory-based morphological analysis, *Proceedings of the 37th Annual Meeting of the ACL*, Morgan Kaufmann, San Francisco, CA, pp. 285–292.

Van der Beek, L., Bouma, G., Malouf, R. and Van Noord, G.(2001), The alpino dependency treebank, *Selected Papers from the Twelfth Computational Linguistics in the Netherlands Meeting, CLIN-2001*, Rodopi, Amsterdam.

Van Eynde, F.(2004), Part of speech tagging en lemmatisering van het Corpus Gesproken Nederlands, *Technical report*, Centrum voor Computerlinguïstiek, K.U. Leuven.

Van Halteren, H., Zavrel, J. and Daelemans, W.(2001), Improving accuracy in word class tagging through combination of machine learning systems, *Computational Linguistics* **27**(2), 199–230.

14

Radio Oranje: Enhanced Access to a Historical Spoken Word Collection

Laurens van der Werff, Willemijn Heeren, Roeland Ordelman, and Franciska de Jong
University of Twente

Abstract

Access to historical audio collections is typically very restricted: content is often only available on physical (analog) media and the metadata is usually limited to keywords, giving access at the level of relatively large fragments, e.g., an entire tape. Many spoken word heritage collections are now being digitized, which allows the introduction of more advanced search technology. This paper presents an approach that supports online access and search for recordings of historical speeches. A demonstrator has been built, based on the so-called Radio Oranje collection, which contains radio speeches by the Dutch Queen Wilhelmina that were broadcast during World War II. The audio has been aligned with its original 1940s manual transcriptions to create a time-stamped index that enables the speeches to be searched at the word level. Results are presented together with related photos from an external database.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

14.1 Introduction

At present, audio(visual) collections from the cultural heritage (CH) domain are at risk of becoming inaccessible, because (i) both the analog data carriers they are stored on are deteriorating and corresponding playback devices are becoming obsolete, and (ii) the materials are insufficiently disclosed for fast and easy access. In this paper we present a demonstrator for online access to a historical audio collection. The technical approach is based on a combination of speech processing and interaction design, and it has been applied to the collection of radio speeches that Queen Wilhelmina (1880-1962) addressed to the Dutch people during World War II – referred to as the 'Radio Oranje collection'. The speeches were broadcast via Radio Oranje, a radio channel set up in London, England, to inform the Dutch people in occupied areas. This demonstrator is an example of how indexing and access to audiovisual collections from the CH domain could be organized to overcome the limitations of traditional indexing methods for A/V material.

Preservation issues have been taken up in retrospective digitization projects for historic audio(visual) collections such as the EU IST PrestoSpace[1] project and the Dutch Beelden Voor De Toekomst[2]. In the case of the Radio Oranje collection, most recordings as well as their original 1940s transcripts underwent preservation measures and have recently been digitized by the Netherlands Institute for War Documentation (NIOD)[3] and the Netherlands Institute for Sound and Vision (NIBG) [4]. Without these measures, the Radio Oranje collection could only be accessed by reading the transcripts kept at the NIOD (in Amsterdam) and/or visiting the NIBG (in Hilversum) to obtain copies of the audio files. As collections become available digitally, they can be made accessible and, in principle, searchable via the Web.

To facilitate keyword search, some textual representation of the audiovisual documents is needed. For the kind of content under discussion here, descriptions typically consist of a set of keywords for long stretches of speech, e.g. an entire hour or tape. This type of metadata is not useless, but it is insufficiently specific to support all needs of a researcher: both the lack of precision in the description and the coarse time-resolution of retrieved results make the exploration of A/V documents quite cumbersome. Moreover, for most of the digitized and digital-born audiovisual documents, disclosure based on manual description is not an option, since manual annotation takes one to ten times the duration of a recording.

To improve access to digitized audiovisual collections it is therefore necessary to automatically generate time-stamped textual representations that describe the spoken content with much more precision (i.e. a higher time-resolution) than is the case in current practice. Automatic generation of a detailed index into the audio can be achieved in several ways, depending on the amount of metadata that is available for a collection. The extremes of the metadata dimension are a full

[1] http://www.prestospace.org/
[2] http://www.beeldenvoordetoekomst.nl
[3] http://www.niod.nl/
[4] http://www.beeldengeluid.nl/

manual transcript on one end, and no metadata at all at the other end. In the former case, aligning the transcription to the audio is sufficient for generating an index, in the latter case automatic speech recognition (ASR) can be employed for generating a textual representation of the spoken content.

In contrast to the broadcast news domain, which has been the main area of speech recognition research and benchmarks, speech from the CH domain can contain relatively large amounts of spontaneous speech (in which speakers overlap, hesitate, repeat themselves, etc.) and of speech that was recorded in adverse conditions (e.g. out on the street) or using suboptimal equipment. A number of research projects have aimed to advance ASR and spoken document retrieval specifically for the CH domain. In The National Gallery of the Spoken Word project, the SpeechFind spoken document retrieval system was developed: it automatically generates metadata for audio documents by segmenting the audio and generating ASR transcripts, and also makes the audio searchable through a Web interface (Hansen et al. 2005). The MALACH (Multilingual Access to Large spoken ArCHives) project investigated access to a vast collection of testimonies from Shoah survivors (Byrne et al. 2004). The goal of that project was to advance English and Czech ASR for the oral history domain and to study how recognition can be best incorporated in further processing and retrieval steps (Gustman et al. 2002). In the Netherlands, the CHoral project[5], part of the NWO-CATCH[6] program, investigates technology for indexing and accessing Dutch, historically relevant spoken documents (Ordelman et al. 2006).

In this paper we will describe a framework for improved access to spoken CH-content. More specifically, we will describe the steps taken to improve access to the Radio Oranje collection. In section 14.2 we will focus on the synchronization step, also called alignment, where the 1940s transcripts were used to generate a time-stamped index of the spoken documents. Section 14.3 discusses how this index was exploited to support online search and browsing and how it was used to enhance presentation. The generation of cross-links to present the speeches together with related photos from an external database will also be explained in this section. Remaining issues and future work are discussed in section 14.4.

14.2 Optimizing alignment

Given the poor sound quality of the speeches – the original recordings were made on historical equipment and contain hiss, pops, and scratches – an ASR engine would not be able to generate an adequate transcript. In the case of an alignment task, audio frames are linked to a phonetic representation of a manual transcript using acoustic models from an ASR system. Alignment is much more robust towards mismatches between models and data than ASR. An example of access to a video archive using alignment of manual transcripts can be found in Christel et al. (2006).

[5] http://hmi.ewi.utwente.nl/choral
[6] http://www.nwo.nl/catch

The collection of speeches in the Radio Oranje project have been fully transcribed during the war and therefore alignment could be done for this collection. The data under consideration consisted of 29 speeches by Queen Wilhelmina, with lengths varying between 5 and 19 minutes. All speeches were manually segmented at the sentence level, giving a total of 853 sentence-sized segments with an average length of 15.7 seconds. For evaluation purposes, two full speeches were segmented at the word level yielding 2028 manually aligned word boundaries. The alignment tool from an off-the-shelf multi-mixture Gaussian HMM-based speech recognition engine was used (Pellom 2001), which produces Viterbi optimized word-based alignments.

14.2.1 Experiment I: Acoustic models

In contrast to an ASR system, which generates a hypothesis of *what* was said, an alignment task only has to decide *where* something was said. Traditionally the same acoustic models are used for both alignment and recognition, but this need not automatically lead to the best alignment result.

We first performed an alignment using gender- and speaker-independent acoustic models, optimized for broadcast news (BN) (de Jong et al. 2006). Both triphone (context-dependent) and monophone (context-independent) BN models were used. New acoustic models were trained from the resulting alignments leading to a speaker-dependent acoustic model. This was then used to perform a second iteration for training the final Wilhelmina models. In total, three different acoustic models were evaluated: a triphone BN model, a monophone BN model and a monophone Wilhelmina model. For these experiments, sentence-sized segments were used as input and the resulting alignment was evaluated at the word level.

14.2.2 Experiment II: Segment size

An alignment tool assigns acoustic model states to each of the audio frames, based on the phonemes that are predicted from the transcription. This is done in such a way that the total likelihood of this state sequence, given the audio, is maximized. Due to the complexity of the task it is not feasible to exhaustively explore all possible alignments. In practice, some pruning is applied and the alignment will converge around a local optimum.

An anchor point is a mark in the audio and the transcription that ties two equivalent positions together. A segment can be viewed as the audio fragment between two adjacent anchor points. When more anchor points are provided to the alignment tool, the task of aligning becomes easier and pruning becomes less of an issue. To determine the influence of segment size on alignment quality, experiments were performed in which alignment was done on varying input sizes. The results were evaluated at the word level.

14.2.3 Experiment III: Grapheme-to-phoneme conversion

Alignment between text and speech is not done directly but through a phoneme representation of the text. First the orthographic transcription is converted into a phonetic representation and then a sequence of acoustic models corresponding to these phonemes can be aligned to the audio. The conversion of graphemes to phonemes has been extensively studied in the past, see Strik and Cucchiarini (1999) for a review. Most grapheme-to-phoneme (G2P) conversion tools produce a canonical phonetic transcription based on a background dictionary that is augmented with a rule-based system. Both the background lexicon and the rules are usually based on modern spelling and the corresponding current pronunciation.

In the CH domain, transcriptions can use archaic spelling conventions as was the case with this collection (e.g. *eisch* instead of *eis*, meaning 'demand', and *voorteekenen* instead of *voortekenen*, meaning 'omens'). To investigate the influence of the G2P conversion on alignment performance, three different phonetic versions of the reference texts were produced and compared: (i) a fully automatic G2P version, (ii) a G2P performed on a version of the reference texts after conversion to modern spelling conventions, and (iii) a manually checked phonetic conversion (thus excluding automatic G2P errors from the process).

14.2.4 Results

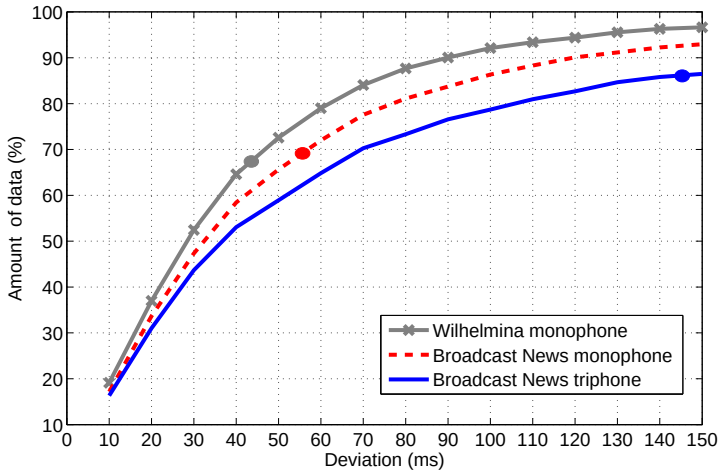


Figure 14.1: Acoustic model performance. For each of the three acoustic models the amount of data complying with a certain amount of deviation from the reference transcript is shown.

Figure 14.1 shows the percentage of word boundaries (vertical axis) that fall

within a certain deviation from the manual reference alignment (horizontal axis). The dots mark the average deviation from the reference. When considering this average deviation, BN monophone acoustic models performed nearly 60% better than traditional BN triphone models on this task. Acoustic models that were specifically trained on these speeches provided an added improvement of almost 20%. The maximum deviation from the reference for all monophone models was less than one second. Regardless of the performance level required, monophone models scored better than triphone models and data-matched models scored better than generic BN models.

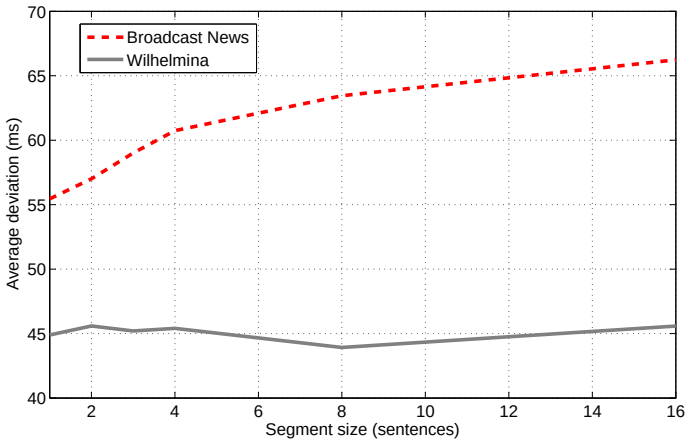


Figure 14.2: Alignment performance as a function of segment size for the BN triphone models and the speaker-adapted Wilhelmina models.

The performance figures that were found for this set are lower than those found in previous studies, see for example Brugnara et al. (1993), where 89% of the aligned phonemes were found within 20 ms of the reference. This stresses the mismatch that exists between the generic BN acoustic models and the historic audio under consideration. Another reason for this difference is that evaluation was done for word-boundaries only, not for phoneme boundaries. This affected the accuracy of the manual placement of reference boundaries, but – as was found in for example Rapp (1995) – it also leads to a slight reduction in overall alignment performance.

Figure 14.2 shows the average alignment error for varying segment sizes. When the data-matched Wilhelmina model is used, segment size seems relatively uncritical. Segments with a length of up to around five minutes do not show a significant reduction in alignment performance when compared to the original sentence-sized segments. Aligning long segments with BN triphone models re-

quired a reduction in pruning that led to a high increase in processing time (>10 times longer).

	Phones altered (%)	Average deviation (ms)
Original spelling	0	55
Modern spelling	1	56
Manual conversion	5	54

Table 14.1: The effect of grapheme-to-phoneme conversion method on alignment performance.

Table 14.1 gives the results for the three types of G2P when the BN monophone acoustic models are used. Not only is the impact of old spelling conventions on G2P quite limited (only 1% of all phonemes is affected), the differences that do exist turn out to be of no consequence for finding the word boundaries. Removing all grapheme-to-phoneme conversion errors from the transcription also shows no significant improvement on alignment performance at the level of word boundaries.

14.2.5 Summary

Overall, alignment performance was more than adequate for this task. The duration of an average syllable lies in the range of 100-300 ms and over 90% of all detected word boundaries were found within 100 ms of the reference. The use of monophone models resulted in better alignment performance than use of traditional triphone models. Segment size was relatively uncritical when the models were well matched to the data. In the case that mismatch between the audio and the models was high, much more processing time was required to obtain acceptable alignment results. Finally, despite the 1940s spelling conventions, there was no impact of grapheme-to-phoneme conversion errors on locating word boundaries.

14.3 Radio Oranje Web interface

On the basis of the alignment, a time-stamped index was created that allows word-level access to the speeches through the Radio Oranje Web interface[7]. The index also facilitated development of additional functionalities for user support. The user experience was enhanced through the generation of cross-links to a topically-related photo collection.

[7]http://hmi.ewi.utwente.nl/choral/radiooranje.html (in Dutch)

14.3.1 Accessing the spoken word documents

For search and browsing, the interface allows entry to the collection at two levels: an entire speech or a speech fragment. It is expected that users will enter their queries in contemporary Dutch spelling, whereas the index contains Dutch in the 1940s spelling. To prevent that words written in the old-fashioned spelling become irretrievable, a dictionary was used to translate terms from user queries into index terms. This dictionary was created manually given the relatively small scale of the task. Boolean retrieval is currently supported and query results are ranked by date, showing the speech's title, broadcast date and duration as well as an excerpt of the relevant sentence fragment. If the framework is used for larger CH collections, more advanced (ranked) retrieval techniques should be used.

Once the user selects a particular spoken word document, basic playback options (start-stop-pause) are insufficient for navigation, as linear examination of the fragments from the result list is relatively inefficient. Therefore, more elaborate and dynamic user controls and content visualization options are needed. In earlier research, visual content representations have been developed that for instance indicate speaker turns (e.g. Slaughter et al. (1998)) or the occurrence(s) of query terms in time (e.g. Whittaker et al. (1999), Christel et al. (2006)). Other tools developed for faster browsing allow users to speed up audio playback, since time-compressed speech remains intelligible up to double its original speed, e.g. Hürst et al. (2004).

To offer a proper mix of flexibility and transparency we developed an interactive visualization of the audio content on the basis of the time-stamped index. It shows an overview of the entire speech as well as a zoomed-in view of a 45 s window around the cursor. The exact positions of highlights, e.g. query terms and sentence boundaries, are shown in both bars. Through this combination of bars it gives a clear overview of the document as well as detailed information on the fragment that is currently being played. This combination is new. Furthermore, the visualization is interactive: clicking any point on either bar will restart the audio at that point in time, which allows the user to quickly browse through the spoken document.

During audio playback, users prefer to take control of playback over predetermined play durations, since restricted playback may stop at unpredictable places (Whittaker et al. 1998). Another issue that may be encountered during playback is that query terms occur right at the beginning of the retrieved fragment. The relevant term may be played before users are well-aware of it. In the W.F. Hermans system[8] this problem was overcome by enabling the user to select the size of the fragment's context (Huijbregts et al. 2005). In the current interface we chose to add an extra button for restarting the fragment from the original entry point.

The second functionality that was added to support users during playback was subtitling. This highlights the word being spoken and shows the query terms in a contrasting color. Subtitling was added to aid intelligibility given the sometimes poor audio quality and the old-fashioned, formal language use encountered in the speeches of Queen Wilhelmina.

[8] http://www.willemfrederikhermans.nl/multimedia/

14.3.2 Cross-media linking

In the CH domain, the ongoing digitization of historical texts, images, pamphlets, photos, audiovisual materials etc. makes it possible to (i) automatically identify links between documents from a variety of modalities and/or collections and (ii) present related documents in one multimedia presentation. These possibilities create new opportunities for comparative research in for instance the historical domain and for the presentation of documents from audiovisual archives for educational purposes. In cross-media linking, content from different media types is associated. This is done by linking the semantic representations from each media type either directly or through, for instance, a thesaurus or ontology. An example is the cross-media browser Infolink that combines broadcast news videos with data from a historical video archive and textual information from a newspaper corpus (Morang et al. 2005).

In our demonstrator, spoken word fragments and photographic material on the same topic, i.e. World War II, were linked. The photographic material was taken from a collection of over 55,000 photos maintained by the NIOD: the photos are partly from the same period as the Radio Oranje broadcasts. However, since unrestricted access to the photo database with elaborate descriptions was not obtained, fully automatic search could not be investigated. The restricted metadata that was available consisted of a few keywords per photo. These keywords were automatically extracted from the online catalog for the photo collection.

Searching and browsing functionality were developed for the spoken content, and as a pre-processing step sets of photos were semi-automatically linked to the speeches. Since the Radio Oranje collection is characterized by a very formal and metaphorical speaking style, it was not possible to automatically match the spoken content to the keywords from the photo database. Therefore, semantic representations for the speeches were generated by manually assigning one or more keywords from the photo thesaurus to each speech. This was done on the basis of its title and global content. The 29 speeches were described by (combinations) of 18 keywords such as Liberation, May 1940, Christmas or Netherlands Indies. These keywords defined photo sets ranging in size from 2 to 200. A number of keywords that were relevant to the entire collection was selected as a default set of photos when speech-level sets are too small, for example: Illegal Press & Radio, Queen Wilhelmina and Dutch Street Scenes.

While the audio is being played, photos that relate to the topic are shown with a refresh rate of ten seconds. Figure 14.3 shows the resulting multimedia presentation: during audio playback the information visualization, subtitling and topically-related photos are shown.

14.4 Discussion and future work

In this paper we have presented the Radio Oranje demonstrator, which is an instantiation of the framework for enhanced spoken word access developed as part of the CHoral project. It shows how access to audiovisual databases from the CH do-

Figure 14.3: Screen shot of the playback interface showing a related photo, subtitling and the interactive browsing bars.

main can be changed using currently available technology for automatic indexing, information retrieval and content visualization.

With respect to the alignment results presented in section 14.2.4, the improved performance from using monophone vs triphone models was in line with expectations (van Santen and Sproat 1999). Although there are some techniques available to improve alignment, such as systematic bias removal (Dines et al. 2002) or spectral boundary correction (Kim and Concie 2002), these were not deemed necessary for the development of this particular system.

To support users during audio browsing and to make their searches as efficient and satisfactory as possible, we developed the information visualization component presented in section 14.3.1. It enables the user to quickly estimate the location of the most important regions within a document given his/her query. In the present demonstrator system, those regions truly contain the query terms given the accurate transcripts. Even if the transcripts were not fully accurate (due to ASR errors for example), the user is expected to be much faster in judging a fragment's relevance using this visualization than without any information on the location of highlights or with less specific information visualizations, see e.g. Whittaker et al. (1999) and Hearst (1995). Future work in the CHoral project will determine how users can be supported even better during retrieval of historical spoken documents.

Another issue for future research concerns semantics. The semantic gap, i.e. the fact that the match between the words spoken and the topic that is being talked about is only partial, should be investigated further. Since manual annotation of high-level semantic information is too (time-)costly, automatic extraction might be a feasible approach. Keywords should ideally be limited to a controlled vocabulary to enable cross-linking with other collections and media. Mapping the terms in the transcription to this vocabulary can be done using a thesaurus- or ontology-type approach as in Wordnet (Fellbaum 1998). Moreover, audiovisual documents on

specific periods or events in history – such as World War II – require the addition of expert knowledge for successfully matching user queries. Words get specific connotations in the context of certain historical periods or events (e.g., euphemisms) that cannot be solved by standard solutions such as document or query expansion using synonyms, hyponyms and hypernyms. Such mappings can – for now – only be provided through manual effort.

In sum, the framework for enhanced spoken word access will be developed further within the CHoral project in order to enable widespread use of Dutch historical spoken word documents in research, education and content production.

Acknowledgements

The research reported on here was funded by the NWO project CHoral, part of CATCH, and supported by the research program MultimediaN (http://www.multimedian.nl). MultimediaN is sponsored by the Dutch government under contract BSIK 03-31.

References

Brugnara, F., Falavigna, D. and Omologo, M.(1993), Automatic segmentation and labeling of speech based on Hidden Markov Models, *Speech Communication* **12**(4), 357–370.

Byrne, W., D.Doermann, Franz, M., Gustman, S., Hajic, J., Oard, D., Picheny, M., Psutka, J., Ramabhadran, B., Soergel, D., Ward, T. and Zhu, W.-J.(2004), Automatic Recognition of Spontaneous Speech for Access to Multilingual Oral History Archives, *IEEE Trans. Speech Audio Proc.*

Christel, M., Richardson, J. and Wactlar, H.(2006), Facilitating access to large digital oral history archives through Informedia technologies, *Proceedings of JCDL '06*, pp. 194–195.

de Jong, F., Ordelman, R. and Huijbregts, M.(2006), Automated speech and audio analysis for semantic access to multimedia, *in* Y. Avrithis, Y. Kompatsiaris, S. Staab and N. O'Connor (eds), *Proceedings of the First International Conference on Semantic and Digital Media Technologies, SAMT 2006*, Vol. 4306 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin, pp. 226–240. ISBN=3-540-49335-2.

Dines, J., Sridharan, S. and Moody, M.(2002), Automatic speech segmentation with hmm, *Proceedings of the 9th Australian Conference on Speech Science and Technology*.

Fellbaum, C. (ed.)(1998), *Wordnet. An electronic lexical database*, MIT Press, Cambridge, MA.

Gustman, S., Soergel, D., Oard, D., Byrne, W., Picheny, M., Ramabhadran, B. and Greenberg, D.(2002), Supporting Access to Large Digital Oral History Archives, *Proceedings of the Joint Conference on Digital Libraries*, pp. 18–27.

Hansen, J., Huang, R., Zhou, B., Deadle, M., Deller, J., Gurijala, A., Kurimo, M. and Angkititrakul, P.(2005), SpeechFind: Advances in spoken document retrieval for a national gallery of the spoken word, *IEEE Transactions on Speech and Audio Processing* **13**(5), 712–730.

Hearst, M. A.(1995), TileBars: Visualization of Term Distribution Information in Full Text Information Access, *Proceedings of the Conference on Human Factors in Computing Systems, CHI'95*.

Huijbregts, M., Ordelman, R. and de Jong, F.(2005), A Spoken Document Retrieval Application in the Oral History Domain, *Proceedings of 10th international conference Speech and Computer, Patras, Greece (SPECOM 2005)*, 2, University of Patras, Wire Communications Laboratory Moscow State Linguistics University, pp. 699–702. ISBN=5-7452-0110-X.

Hürst, W., Lauer, T. and Götz, G.(2004), An elastic audio slider for interactive speech skimming, *Proceedings of NordCHI '04*.

Kim, Y.-J. and Concie, A.(2002), Automatic segmentation combining an hmm-based approach and spectral boundary correction, *Proceedings of ICSLP 2002*, pp. 145–148.

Morang, J., Ordelman, R., de Jong, F. and van Hessen, A.(2005), Infolink: analysis of Dutch broadcast news and cross-media browsing, *Proceedings of IEEE International Conference on Multimedia and Expo, ICME 2005*, pp. 1582–1585.

Ordelman, R., de Jong, F. and Heeren, W.(2006), Exploration of Audiovisual Heritage Using Audio Indexing Technology, *Proc. of the 1st workshop on Intelligent Technologies for Cultural Heritage Exploitation*, pp. 36–39.

Pellom, B.(2001), SONIC: The University of Colorado Continuous Speech Recognizer. Technical Report TR-CSLR-2001-01, University of Colorado.

Rapp, S.(1995), Automatic phonemic transcription and linguistic annotation from known text with Hidden Markov Models / An Aligner for German, *Proceedings of ELSNET goes east and IMACS Workshop "Integration of Language and Speech in Academia and Industry"*.

Slaughter, L., Oard, D. W., Warnick, V. L., Harding, J. L. and Wilkerson, G. J.(1998), A graphical Interface for Speech-Based Retrieval, *ACM DL*, pp. 305–306.

Strik, H. and Cucchiarini, C.(1999), Modeling pronunciation variation for ASR: A survey of the literature, *Speech Communication* **29**, 225–246.

van Santen, J. and Sproat, R.(1999), High-accuracy automatic segmentation, *Proceedings of EuroSpeech99*.

Whittaker, S., Hirschberg, J. and Nakatani, C.(1998), Play it again: a study of the factors underlying speech browsing behavior, *Proceedings of CHI 1998*.

Whittaker, S., Hirschberg, J., Choi, J., Hindle, D., Pereira, F. C. N. and Singhal, A.(1999), SCAN: Designing and Evaluating User Interfaces to Support Retrieval From Speech Archives, *Research and Development in Information Retrieval*, pp. 26–33.

15

Extraction of Dutch definitory contexts for eLearning purposes

Eline Westerhout and Paola Monachesi
Utrecht University

Abstract

The aim of the Language Technology for eLearning project is to facilitate the retrieval, management and distribution of learning material within a Learning Management System by exploiting Natural Language Processing techniques as well as semantic knowledge. One of the functionalities provided by the project is the possibility to create glossaries semi-automatically. Glossaries are derived from the learning objects in order to capture the exact definition which the author of these documents uses. A rule-based approach is employed to identify the relevant lexical and linguistic patterns which underlie the definition. In this paper, we discuss the grammar developed to identify the definitory contexts in the Dutch learning objects and we present the results of the quantitative evaluation.

15.1 Introduction

The aim of the European project Language Technology for eLearning (LT4eL)[1] is to show that the integration of Language Technology based functionalities and Semantic Web techniques will enhance the management, distribution and retrieval of

[1] http://www.lt4el.eu.

Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands
Edited by: Peter Dirix, Ineke Schuurman, Vincent Vandeghinste, and Frank Van Eynde.

the learning material within Learning Management Systems (LMS). The functionalities are being developed for eight languages represented in our consortium that is Bulgarian, Czech, Dutch, English, German, Polish, Portuguese and Romanian (Monachesi et al. 2006b), (Monachesi et al. 2006a).

Language Technology resources and tools, such as corpora and taggers which have been produced in the context of other projects are employed in the development of new functionalities that will allow the semi-automatic generation of metadata for the description of learning objects in an LMS: to this end, a keyword extractor is being developed (Lemnitzer et al. 2007).

Furthermore, the project will integrate the use of ontologies, a key element in the Semantic Web architecture, to structure and retrieve the learning material within the LMS. An ontology of 1000 concepts for the domain of computer science for non-experts and eLearning has been developed as well as an English vocabulary and English annotated learning objects. The ontology should facilitate the multilingual retrieval of learning objects.

Another objective of the project is the semi-automatic construction of glossaries which will be built on the basis of the definitory contexts which are presented in the learning objects themselves in order to capture the exact definition which the author of these documents uses. This definition in many cases overrides a more general definition of the term.

In the project, definitory contexts are identified in a bottom-up manner. First, a substantial amount of definitions are selected and annotated manually in the learning objects which are the asset of this project. From these examples, grammars with the complexity of regular languages are abstracted (cf. Muresan and Klavans (2002) for a similar approach). These language-specific grammars are applied to a test set from the same language in order to estimate their coverage.

In this paper, we focus on the definitory contexts attested in the Dutch learning objects and the grammar necessary to identify them. As a basis for the extraction and annotation, we use linguistically annotated learning material which has been converted into XML. This process is discussed in section 15.2. Our approach to the detection and extraction of definitory contexts is rule-based. The patterns covered by our grammar are discussed in section 15.3 and 15.4 while the grammar is presented in section 15.5. Section 15.6 deals with the results we have obtained with the current version of the grammar. In section 15.7, we compare our methodology with other approaches while section 15.8 contains our conclusions and suggestions for future work.

15.2 The corpus

The learning material which constitutes our corpus from which definitions are extracted, can have different formats, such as HTML, PDF or DOC. Figure 15.1 illustrates the conversion process from the original file into the final XML output which conforms to the LT4ELAna DTD. This DTD has been derived from the XCES DTD for linguistically annotated corpora (Ide and Suderman 2002). For our purposes, the XCES DTD has been enriched with elements which are relevant

for our project and contains – besides the content of the original files (that is, information about layout and the text itself) – the possibility to encode information about part-of-speech, morphosyntactic features and lemmas. This information is used for the extraction of keywords and the detection of definitory contexts.

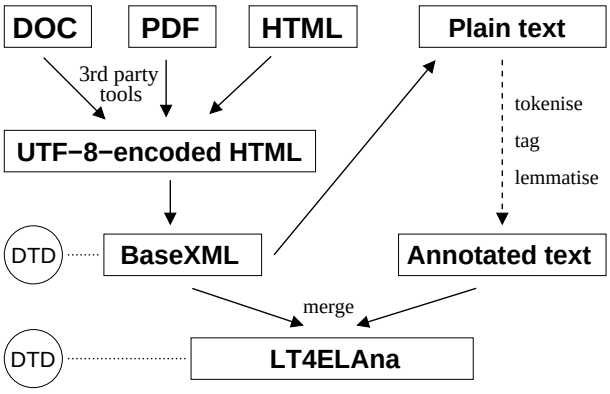


Figure 15.1: Data flow for the processing of learning objects.

The Wotan tagger (Daelemans et al. 1996) has been used for annotating the Dutch documents with part-of-speech information and morphosyntactic features whereas the CGN lemmatizer (Bosch and Daelemans 1999) was used for the lemmatization.

Figure 15.2 presents an example sentence in the LT4ELAna format. The *id* attribute is a unique identifier for each word, the *base* attribute contains the lemma of the word, the *ctag* attribute is related to the part-of-speech tag and the *msd* attribute gives the morpho syntactic information. The layout information is stored in the *rend* attribute. The rules of the grammar for the extraction of the definitory context are based on the information encoded in the LT4ELAna format.

```
<s id="s150">
  ...
  <tok id="t2254" class="word" base="het" ctag="Art"
    msd="bep,onzijd,neut">het</tok>
  <tok id="t2255" class="word" rend="b"
    base="eLearning-actieplan" ctag="N" msd="soort,ev,neut">
    eLearning-actieplan</tok>
  <tok id="t2256" class="punc" rend="b" base="." ctag="Punc"
    msd="punt">.</tok>
</s>
```

Figure 15.2: Part of a sentence in LT4ELAna format

15.3 The use of definitory contexts

Research on the detection and identification of definitory contexts has been pursued mainly in the context of question answering systems, where finding answers to definitory questions is a particularly difficult problem (cf. among others Miliaraki and Androutsopoulos (2004), Blair-Goldensohn et al. (2004) and Fahmi and Bouma (2006)). Very often pattern matching techniques are used to detect definitions such as the one exemplified below:

(1) Een vette letter is een letter die zwarter wordt afgedrukt dan de
a bold character is a character that blacker is printed than the
andere letters.
other characters
'A bold character is a character which is printed darker than the other characters'.

Definitory contexts are expected to contain at least three elements: (1) the definiendum, that is the element that is defined (i.e. *een vette letter*), (2) the connector, which indicates the relation with the third element (i.e. *is*) and (3) the definiens, that is the definition of the definiendum (*een letter die zwarter wordt afgedrukt dan de andere letters*) (Walter and Pinkal 2006, Fahmi and Bouma 2006). The number of patterns distinguished by the various systems differs largely. The documents used to extract definitory contexts are usually dictionaries or encyclopedias, which contain well structured definitions.

The LT4eL project is quite innovative with respect to the research in this area because it has adopted well known techniques to extract definitions and provided a totally new application: in the field of eLearning, identifying definitory contexts is relevant for the construction and maintenance of glossaries (Monachesi et al. 2006b). Furthermore, within our project the extracted definitions are employed in the construction of a domain ontology.

Glossaries are an important kind of secondary index to a text. They can be seen as small lexical resources which support the reader in decoding the text and understanding the central concepts which are conveyed. Since a glossary can be built on the definitory contexts which are present in the learning objects themselves, the advantages for the learning process are obvious: the learner accesses the appropriate definition which is the one used by the author of the learning object, which can in certain cases be different from the general definition of the term that could be found in a dictionary. For example, when we encounter the word 'enter' in a tutorial about Word, it will not have the meaning given by the Merriam Webster dictionary: 'to go or come in'. Instead, it will most times stand for the enter key and therefore have a completely different definition, that is: 'Also known as a return key, the enter key is used to return a cursor to the next line or execute a command or operation. It is common for most standard keyboards to have two enter or return keys, one on the keyboard and another on the numeric keypad'.

15.4 Types of definitory contexts

In order to identify the typology of definitions attested in our corpus, we have manually extracted 303 definitory contexts from our learning objects and grouped them according to the connector used. It should be mentioned that the collection of Dutch learning objects comprises 77 files within three different domains: computer science for non-experts (e.g. manuals on software programs), eLearning and the Pulman documents which deal with digitization. The average number of tokens per file is 6568 and the average number of types is 463.

The creation of the grammar has been done on the basis of the patterns found in 21 files. These 21 files contain 303 definitory contexts. We call this the training corpus. It should be noted that we are not using machine learning techniques yet, the files have not been used for training in the sense of training a classifier but only to identify the most common patterns. The test corpus consists of 14 files and has only been used for evaluating the grammar. It contains 159 definitory contexts

We distinguish three elements in definitory contexts (i.e. the definiendum, the connector and the definiens) in our approach. According to the patterns, the definitory contexts were classified into five groups:

1. Definitory contexts in which a form of the verb *zijn* ('to be') is used as connector verb;

```
Gnuplot is een programma om grafieken te maken .
'Gnuplot is a program for drawing graphs'
```

2. definitory contexts in which other verbs are used as connector (e.g. *betekenen* ('to mean'), *wordt ... genoemd* ('is called'), *wordt gebruikt om* ('is used to'));

```
E-learning omvat hulpmiddelen en toepassingen die via het
internet beschikbaar zijn en creatieve mogelijkheden
bieden om de leerervaring te verbeteren .
'eLearning comprises resources and application that are
available via the internet and provide creative possibilities
to improve the learning experience'
```

3. definitory contexts having specific punctuation features (e.g. *:*, *(..)*);

```
Passen: plastic kaarten voorzien van een magnetische strip,
die door een gleuf gehaald worden, waardoor de gebruiker
zich kan identificeren en toegang krijgt tot bepaalde
faciliteiten.
'Passes: plastic cards equipped with a magnetic strip, that
can be swiped through a card reader, by means of which the
identity of the user can be verified and the user gets
access to certain facilities'
```

4. definitory contexts in which the layout plays an important role (e.g. in tables, defined term in margin, defined term in heading);

```
RABE
Een samenwerkingsverband van een aantal Duitse bibliotheken,
die gezamenlijk een Internet inlichtingen dienst bieden,
gevestigd bij de gemeenschappelijke catalogus, HBZ, in
Keulen.
'RABE,
Cooperation of a number of German libraries, that
together provide an Internet information service, residing
at the common catalogue, HBZ, in Cologne'
```

5. definitory contexts in which relative and demonstrative pronouns (e.g. *dit* ('this'), *dat* ('that'), *deze* ('these')) and words like *hiermee* ('with this'), *hierdoor* ('because of this') are used to point back to an earlier used defined term. The definition of the term then follows after the pronoun, so these are often multisentence definitory contexts.

```
Dedicated readers.
Dit zijn speciale apparaten, ontwikkeld met het exclusieve
doel e-boeken te kunnen lezen.
'Dedicated readers.
These are special devices, developed with the exclusive
goal to make it possible to read e-books.'
```

Some definitions can be classified in more than one category. For these cases, we have chosen the category that was most important for the identification of the pattern. For example, in the last example, both the layout and the pronoun 'Dit' can be used as clues. We classified it as a pronoun definition, because 'dit' gives a stronger clue than the layout does. Table 15.1 shows how the definitory contexts are divided over the 5 types. From this table we can see that the definitions with a form of the verb *zijn* ('to be') as connector verb account for respectively 27.7 % and 38.4 % of the definitions and that in both the test and the training corpus around 40 % of the definitory contexts does not have a verb as main indicator.

	Training corpus	Test corpus
Type 1	84	61
Type 2	99	41
Type 3	46	13
Type 4	7	1
Type 5	46	27
Other patterns	48	31
# sentences	330	174
# definitory contexts	303	159

Table 15.1: Division of the definitory contexts into types

Although there are 303 definitions in the training corpus, we have more sentences, because definitory contexts have been identified which consist of more than

one sentence (i.e. often two sentences are present). In most multisentence definitory contexts, one of the sentences contains only the defined term and no explanation of its meaning. These sentences in which only the defined term is mentioned do not meet our definition of a definitory context, and are therefore not identified by our grammar and also not mentioned in table 15.1. This is for example the case in the multisentence definitory context below:

```
Een gebruiker kan meer dan een programma tegelijkertijd
draaien. Dit wordt multi-tasking genoemd.
'A user can run more than one program at a time.
This is called multi-tasking.'
```

We leave out the sentences which contain only the defined term when we evaluate the performance of the grammar. As a consequence, we have only 27 multisentence definitory contexts left in the training corpus and 15 in the test corpus. The second part of the multisentence definitory contexts fits either in the fifth definition category or does not have a definitory context pattern. For these cases, both sentences give information on the meaning of the term defined.

```
TEX is een computerprogramma van Donald E. Knuth.
Het is speciaal ontworpen voor het zetten en drukken
van mathematische teksten en formules.
'TEX is a computer program developed by Donald E. Knuth.
It has been designed for setting and printing mathematical
texts and formulas.'
```

As already mentioned, most approaches to definition extraction use dictionaries or encyclopedias as corpus. This is not the case of our project in which the learning objects which constitute our corpus are mainly manuals and articles. As a consequence we have identified a variety of definitory context patterns which have not been taken into consideration in previous studies. This is the case for type 3, 4 and 5 patterns which are less common in dictionaries and encyclopedias. For some of these definitions, it is even not immediately clear that they are definitory contexts. The context of the patterns then determines whether or not we have to do with a definition. The type 3, 4 and 5 patterns make our work challenging and innovative.

15.5 The grammar

As already mentioned, in the LT4eL project, we have adopted a rule-based approach to the extraction of definitory contexts. Because of the variety of patterns present in our learning objects, we believe this is the best approach to use. Previous research has shown that grammars which match the syntactic structures of the definitory contexts are the most successful approaches if deep syntactic and semantic analysis of texts is not available (Muresan and Klavans 2002, Liu et al. 2003).

Therefore, we have developed a Dutch grammar in order to extract the definitory context patterns. The XML transducer *lxtransduce* developed by Tobin (2005) is used to match the grammar against files in the LT4eLAna format. Lxtransduce is an XML transducer, especially intended for use in NLP applications. It supplies a format for the development of grammars which are matched against either pure

text or XML documents. The grammars must be XML documents which conform to a DTD (lxtransduce.dtd, which is part of the software). In each grammar, there is one "main" rule which calls other rules by referring to them. The XPath-based rules are matched against elements in the input document. When a match is found, a corresponding rewrite is done.

The grammar contains rules that match the grammatical patterns described in the previous section. The rules have been written on the basis of the 303 manually selected definitory contexts. At the moment, we focus on the extraction of patterns in which verbs are used as connector (type 1 and type 2). For type 3, we can extract the patterns with the colon as connector and the patterns between brackets. For type 5, we can extract patterns in which words like 'hiermee' ('with this') are used and definitions starting with 'dit' ('this'). Type 4 has not been implemented yet.

The grammar consists of four parts. In the first part, the part-of-speech information is used to make rules for matching separate words (e.g. verbs, nouns, adverbs). The second part consists of rules to match chunks (e.g. noun phrases, prepositional phrases). We did not use a chunker, because we want to be able to put restrictions on the chunks. The third part contains rules for matching and marking the defined terms and in the last part the pieces are put together and the complete definitory contexts are matched. The rules were made as general as possible to prevent overfitting to our training corpus.

Figure 15.3 shows one of the rules described in the fourth part, namely the rule for the extraction of definitory contexts in which a form of *to be* ('zijn') is used as connector. The *name* attribute in the element *ref* refers to a previously described rule with this name, so the first element of the rule refers to a rule defined in the third part of the grammar with the name *markedTerm* and matches 'een vette letter'. Thereafter, the verb is matched ('is'). After the verb, a noun phrase follows ('een letter'). The rest of the sentence is matched with the rule 'tok_or_chunk', which identifies the relevant material until the end of the sentence.

15.6 Results

The current grammar is able to detect type 1, type 2, type 3 and type 5 patterns. We have left type 4 (layout patterns) aside, for the moment, due to the low frequency of this pattern which makes the identification of the appropriate rules for its detection a complex task.

We calculated the performance of the grammar for each of the types in terms of precision, recall and F-score. In the evaluation, precision and recall were calculated at two levels: at the token level and at the sentence level, as both ways of the evaluation of definition extraction may be found in the literature. At the token level, precision is understood as the number of tokens simultaneously belonging to a manual definition and an automatically found definition, divided by the number of tokens in automatically found definitions. Correspondingly, recall is the ratio of the number of tokens simultaneously in both definition types to the number of tokens in manual definitions. At the sentence level, a sentence is taken as a manual

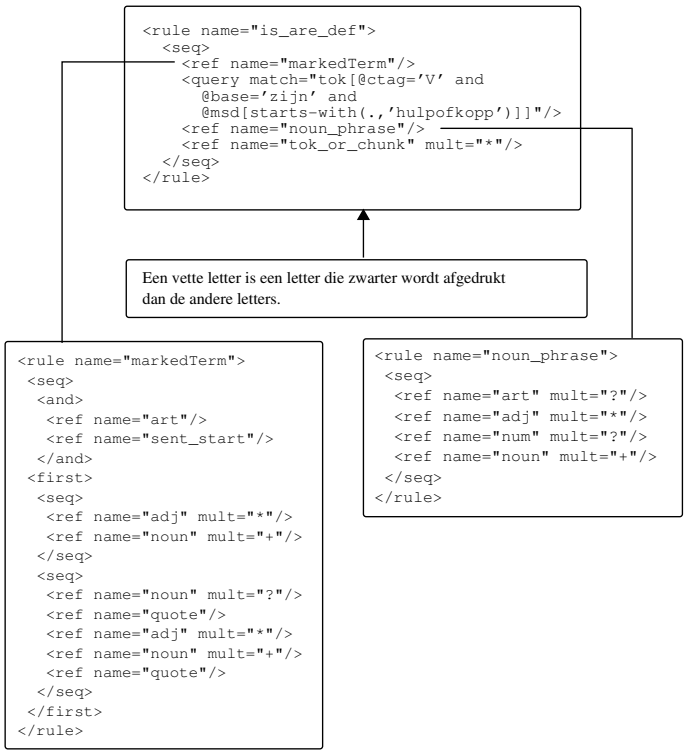


Figure 15.3: Grammar rule for extracting *is*-patterns

or automatic definition sentence if and only if it contains a (part of a), respectively, manual or automatic definition. Given that, precision and recall are calculated in a way analogous to token level precision and recall. It is important to select the appropriate units when measuring precision and recall. We think for our task the sentence is the most appropriate unit and therefore we report the results obtained when using the sentence as a unit (Przepiórkowski et al. 2007).

We did not only calculate the usual F-score, but also the F_2-score. In this score, recall is weighted twice as much as precision[2]. For the task at hand, where recall is more important than precision, the latter measure in which recall is measured seem appropriate (Przepiórkowski et al. 2007). The performance of the grammar has been evaluated for both the training set and the test set.

For type 1 (the *is*-patterns), we had a recall of 73.81, a precision of 22.63 and an F_2-score of 42.08 on the training corpus and a recall of 91.80, a precision of 20.97 and an F_2-score of 43.18 on the test corpus (Table 15.2).

Within the test set, the grammar was able to detect 56 out of 61 definitory

[2] $F_\alpha = (1+\alpha) \cdot (\text{precision} \cdot \text{recall}) / (\alpha \cdot \text{precision} + \text{recall})$. For F_2, $\alpha = 2$

		Precision	Recall	F_1-score	F_2-score
Type 1	training	22.63	73.81	34.64	42.08
	test	20.97	91.80	34.15	43.19
Type 2	training	44.64	75.76	56.18	61.48
	test	25.76	41.46	31.78	34.46
Type 3	training	5.71	54.35	10.33	14.15
	test	2.58	76.92	4.99	7.25
Type 5	training	9.18	41.30	15.02	19.06
	test	6.15	40.74	10.68	14.16

Table 15.2: Performance of the grammar

contexts. For three of the non-detected sentences, the verb 'is' was followed by an adverb or an adverbial used adjective. The other two sentence were not found due to an error of the part-of-speech tagger (e.g. the word 'uitwerken' (elaborating) was tagged as a verb, whereas it is used as a noun in this context). The recall is slightly better for the training set.

The type 2 patterns are those in which a verb different from *zijn* ('is') is used as connector. For the training corpus, recall was 74.76, precision was 44.46 and the F_2-score was 61.48. For the test corpus, both recall and precision were remarkably lower, namely 41.46 and 25.76. The F_2-score on the test corpus was 34.46.

It should be noticed that a number of verbs can be used as connector, such as *betekenen* ('to mean'), *omvatten* ('to comprise'), *bestaan uit* ('to consist of'), *wordt gedefinieerd als* ('can be defined as'). However, there are also verbs that are used within definitory contexts that are normally not used as connector, such as the verb *voorkomen* ('prevent').

(2) Een vaste spatie voorkomt dat een regel tussen twee woorden
A non-breaking space prevents that a line between two words
wordt afgebroken.
is splitted
'A non-breaking space prevents a line from being splitted between two words'.

Whereas not everybody will consider this as a definition, they probably will consider the next sentence, which contains the same information, as a definition:

(3) Een vaste spatie is een spatie die voorkomt dat een regel
A non-breaking space is a space that prevents that a line
tussen twee woorden wordt afgebroken.
between two words is splitted
'A non-breaking space is a space that prevents a line from being splitted between two words'.

Because of the diversity of possible type 2 patterns, the recall score for type 2 is lower than the recall score for type 1. The precision is higher for type 2, because the patterns in which connector verbs different from a form of 'to be' are used, are less common in non-definitory contexts.

The third type of patterns comprises the patterns in which there is a punctuation character indicating that the sentence is a definition (e.g. a colon or brackets). The main problem for the identification of this type of definition is that it also occurs very often in non-definitory contexts. The precision is therefore very low (5.71 on training corpus and 2.58 on the test corpus). Recall is higher for the test corpus than it is for the training corpus (76.92 and 54.35 respectively), but the F-score is higher for the training corpus.

Within the type 5 patterns, two groups can be distinguished. The first group contains definitions starting with *dit* and the second group contains definitions starting with words like *hiermee*. The first type of definitions has roughly the same pattern as the type 2 definitions, whereas within the second type other patterns are used. All scores are higher for the training corpus: precision is 41.30 on the training corpus and 40.74 on the test corpus. Recall is respectively 9.18 and 6.15, and the F-scores are also higher for the training corpus.

In our project, we have a broad definition of what a definitory context is. Our learning objects present us with patterns that are often not attested in encyclopedias and dictionaries. Around 60 % of our patterns are standard definition patterns (i.e. definitions in which a verb is used as connector). However, this implies that we also have around 40 % non-standard patterns (that is, patterns of type 3, 4 or 5). Because of the variety of patterns attested in our corpus, we believe that a rule-based approach is the most appropriate for our task.

In the analysis of our results, we should take into account that there are several definition patterns that can also occur in non-definitory contexts. This is often the case for *to be* patterns and punctuation patterns and this has obviously a negative influence on the precision scores, as shown by the example below, which has the structure of a definition but it is obviously not a definition.

```
De stad is een belangrijke havenstad aan de Middellandse Zee.
'The city is an important port in the Mediterranean.'
```

Even though we used a state-of-the-art tagger (Bosch and Daelemans 1999), some of the definitory contexts were not found due to a tagger error. Most times, errors are nouns tagged as verbs (e.g. 'leren' in 'Levenslang leren' ('learning' in 'lifelong learning') or English words or commands (e.g. 'sort' referring to the Unix command 'sort' is tagged as verb). For the *zijn*-pattern, 27.3 % of the definitory contexts (6 definitory contexts) that were not found by the grammar, were not detected due to errors of the part-of-speech tagger.

15.6.1 Interannotator agreement

Because it is not always clear whether a sentence is a definitory context or not, it would be relevant to have more annotators expressing their judgments. We

should let them analyze both the manually annotated definitory contexts to see whether they really are definitory contexts and the definitory contexts extracted by the grammar which were not marked by the annotator to check whether some of these can also be accepted as definitory contexts. These judgments could lead to the deletion and addition of some definitory contexts, which would result in an improvement of both precision and recall.

More generally, it would be relevant to identify the interannotator agreement in the annotation of definitions within our corpus and therefore we have carried out a small experiment to this end (Muresan and Klavans 2002). One of our texts was provided to three other persons which were asked to annotate the definitions and their headwords in this text. In total, 87 different sentences were marked as definitory context by the 4 annotators, 52 of which were unique.

We measured the interannotator agreement using Cohens kappa (κ) and several adapted versions of it (Table 15.3). Cohens kappa is the standard version of kappa. It assumes that the scores are equally divided over the categories. However, we have a large difference between the number of definitions and non-definitions in a text. Therefore, we also used another statistical measure in which this is taken into account. This score, the PABAK score (prevalence-adjusted bias-adjusted kappa), accounts for prevalence and bias of the data (Byrt et al. 1993). The True Skill Statistic (TSS) can be used when one of the annotators is considered to be an expert (Allouche et al. 2006). The annotation of the expert is then taken as model and the definitions marked by the other annotators are compared to this. In this case, we used our own annotation as expert annotation (annotator 4) and compared the results of the other annotators to these definitions.

Annotators	Cohens κ	PABAK	TSS
1 + 2	0.26	0.4	
1 + 3	0.27	0.43	
1 + 4	0.24	0.45	0.58
2 + 3	0.37	0.6	
2 + 4	0.42	0.69	0.77
3 + 4	0.42	0.74	0.62

Table 15.3: Interannotator agreement

The experiment with more annotators shows that the agreement between different annotators is not very high when definitions have to be annotated. From the fact that 87 different sentences were marked as definitory context by 4 annotators from which only 35 were marked by more then one person, we can already see that it is not easy to distinguish definitory contexts. The statistics in table 15.3 support this intuitive thought: both the Cohens κ score and the PABAK score show that the agreement between the different annotators is not very high. Although the agreement is better when we consider our own annotation as expert annotation and compare the others to this (TSS-scores) the agreement is higher, it is still not very

high.

For measuring the interannotator agreement, it should be investigated which is the best statistical method to evaluate interannotator agreement for our purposes. Besides, the experiment should be repeated with a larger set of documents to make it possible to draw stronger conclusions.

15.7 Related work

Research on the detection of definitory contexts has been pursued mainly in the context of question-answering tasks. The answers to 'What is'-questions are usually definitions of concepts. A common approach in this field is to search the corpus for sentences consisting of a subject, a copular verb and a predicative phrase. If the concept matches the subject, the predicative phrase is returned as answer. However, although the recall is high for this approach, the precision is often low, because there are many sentences which have the relevant syntactic form but are not definitions (Tjong Kim Sang et al. 2005). We encountered this problem within our approach for the patterns with a form of *zijn* ('to be') as connector. Fahmi and Bouma (2006) tried to solve this problem by applying machine learning techniques on the potential definitory contexts they extracted. They succeeded to improve the precision with 16.3 %. For this reason, we plan to adopt machine learning techniques to improve our results.

Within the German HyTex project (Storrer and Wellinghof 2006), 19 definitor verbs were distinguished on the basis of 174 manually extracted definitory contexts. Sentences in which one of these verbs was used were extracted. The results were calculated for each of the different definitors. They differed highly for the 19 verbs and depended also on the number of times the pattern was used. For the precision, the most problematic verb was the verb *sein* ('to be'), for which a precision of only 31 % was achieved. This is comparable to our precision score for this type of patterns. The recall was worst for the verb *nennen* ('to call') (20 %).

The DEFINDER system (Muresan and Klavans 2002) combines shallow natural language processing with deep grammatical analysis to identify and extract definitions and the terms they define from on-line consumer health literature. Four persons were provided with a set of nine articles, and were asked to annotate the definitions and their headwords in text. The gold-standard against which the system was compared, was determined by the set of definitions marked up by at least 3 out of the 4 subjects and consisted of 53 definitions. Nearly 60% of the definitions are introduced by a limited set of text markers '–', '()'), the other 40% being identified by more complex linguistic phenomena (anaphora, apposition, conjoined definitions). DEFINDER identified 40 out of the 53 definitions obtaining 86.95% precision and 75.47% recall. We used the same approach for one of our files to investigate whether this would lead to a different set of definitions. Because we used only one text, the differences for type 1 and type 2 were small compared to the results obtained by comparisaon to the set of definitions annotated by one person. It is difficult to compare our results to the DEFINDER results, because they use more structured texts.

15.8 Conclusions and future work

One of the functionalities developed within the LT4eL project is the possibility to derive glossaries semi-automatically on the basis of the definitory contexts identified within the learning objects.

A rule-based approach is employed to identify the definitory contexts. The current grammar is able to identify most types of definitory contexts and we obtain an acceptable recall while precision should be improved. However, due to the embedding of this functionality within an eLearning context in which human intervention is foreseen, the results are quite good.

At the moment, we are working on the improvement of the results at several levels.

First, we will investigate to which extent machine learning techniques can be used to improve the results and adopt an approach similar to Fahmi and Bouma (2006) to filter out unwanted results. More generally, we will have an identification step in which definitions will be detected on the basis of NLP techniques which will be followed by a filtering step based on machine learning techniques. We believe that we would always need to identify the definitions by means of a grammar, because this is the best approach to identify the relevant patterns and will enable us to to generalize the approach across all the languages involved in our project. Furthermore, we are not aware of machine learning approaches that account for the extraction of definitory contexts of type 3, 4 and 5.

As for the grammar, we will extend it with additional rules to cover also the less frequent patterns. In addition, we will investigate to which extent the grammar can be made more language independent. To this purpose, we are closely cooperating with the German and English grammar developers within the project to see whether the patterns of definitions are similar in closely related languages.

More generally, we wonder whether a quantitative evaluation is the best way to evaluate our results. Due to the variety of patterns attested and the lack of agreement among users about what should be considered a definition, it might be more appropriate to evaluate our grammar also from a qualitative point of view. Given the eLearning context in which we operate, the definitory contexts will be used to develop glossaries that are linked to the various learning objects, it might be thus more relevant to evaluate the degree of satisfaction of the users. These are both the content providers who will exploit this functionality in order to develop glossaries semi-automatically and they can thus select among the proposed definitions those that they consider the most appropriate as well as the learners who thanks to this functionality will have glossaries at their disposal that should facilitate their learning process.

References

Allouche, O., Tsoar, A. and Kadmon, R.(2006), Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (tss), *Jour-*

nal of applied ecology **43**(6), 1223–1232.

Blair-Goldensohn, S., McKeown, K. and Hazen Schlaikjer, A.(2004), *New Directions In Question Answering*, AAAI Press, chapter Answering Definitional Questions: A Hybrid Approach.

Bosch, A. v. d. and Daelemans, W.(1999), Memory-based morphological analysis, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, ACL '99*, pp. 285–292.

Byrt, T., Bishop, J. and Carlin, J.(1993), Bias, prevalence and kappa, *J Clin Epidemiol* **46**(5), 423–429.

Daelemans, W., Zavrel, J., Berck, P. and Gillis, S.(1996), Mbt: A memory-based part of speech tagger generator, *in* E. Ejerhed and I. Dagan (eds), *Proceedings of the Fourth Workshop on Very Large Corpora*, pp. 14–27.

Fahmi, I. and Bouma, G.(2006), Learning to identify definitions using syntactic features, *in* R. Basili and A. Moschitti (eds), *Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications*.

Ide, N. and Suderman, K.(2002), XML Corpus Encoding Standard, document XCES 0.2, *Technical report*, Department of Computer Science, Vassar College, and Equipe Langue et Dialogue, New York, USA and LORIA/CNRS, Vandoeuvre-lés-Nancy, France. See `http://www.cs.vassar.edu/XCES/`.

Lemnitzer, L., Vertan, C., Killing, A., Simov, K., Evans, D., Cristea, D. and Monachesi, P.(2007), Improving the search for learning objects with keywords and ontologies, *Proceedings of ECTEL 2007*.

Liu, B., Chin, C. and Ng, H.(2003), Mining topic-specific concepts and definitions on the web, *Proceedings of WWW-2003*.

Miliaraki, S. and Androutsopoulos, I.(2004), Learning to identify single-snippet answers to definition questions, *Proceedings of COLING 2004*, pp. 1360–1366.

Monachesi, P., Cristea, D., Evans, D., Killing, A., Lemnitzer, L., Simov, K. and Vertan, C.(2006a), Integrating language technology and semantic web techniques in elearning, *Proceedings of ICL 2006*.

Monachesi, P., Lemnitzer, L. and Simov, K.(2006b), Language technology for elearning, *in* W. Nejdl and K. Tochtermann (eds), *Proceedings of EC-TEL 2006*, Springer LNCS, pp. 667–672.

Muresan, S. and Klavans, J.(2002), A method for automatically building and evaluating dictionary resources, *Proceedings of the Language Resources and Evaluation Conference (LREC 2002)*.

Przepiórkowski, A., Degórski, L., Spousta, M., Simov, K., Osenova, P., Lemnitzer, L., Kubon, V. and Wójtowicz, B.(2007), Towards the automatic extraction of definitions in slavic, *Proceedings of BSNLP workshop at ACL*.

Storrer, A. and Wellinghof, S.(2006), Automated detection and annotation of term definitions in German text corpora, *Proceedings of LREC 2006*.

Tjong Kim Sang, E., Bouma, G. and de Rijke, M.(2005), Developing offline strategies for answering medical questions, *in* D. Mollá and J. Vicedo (eds), *Pro-*

ceedings AAAI 2005 Workshop on Question Answering in Restricted Domains.

Tobin, R.(2005), Lxtransduce, a replacement for fsgmatch. See `http://www.ltg.ed.ac.uk/~richard/ltxml2/lxtransduce-manual.html`.

Walter, S. and Pinkal, M.(2006), Automatic extraction of definitions from German court decisions, *Proceedings of the workshop on information extraction beyond the document*, pp. 20–28.

List of Contributors

Lou Boves
Afdeling Taalwetenschap, Radboud Universiteit Nijmegen
Postbus 9103, NL-6500 HD Nijmegen, The Netherlands
l.boves@let.ru.nl

António Branco
Departamento de Informática, Universidade de Lisboa
Campo Grande, P-1749-016 Lisboa, Portugal
antonio.branco@di.fc.ul.pt

Bertjan Busser
ILK Research Group, Universiteit van Tilburg
Postbus 90153, NL-5000 LE Tilburg, The Netherlands
g.j.busser@uvt.nl

Sander Canisius
ILK Research Group, Universiteit van Tilburg
Postbus 90153, NL-5000 LE Tilburg, The Netherlands
s.v.m.canisius@uvt.nl

Francisco Costa
Departamento de Informática, Universidade de Lisboa
Campo Grande, P-1749-016 Lisboa, Portugal
fcosta@di.fc.ul.pt

Walter Daelemans
Centrum voor Nederlandse Taal en Spraak, Universiteit Antwerpen
Campus Drie Eiken, Universiteitsplein 1, B-2610 Wilrijk, Belgium
walter.daelemans@ua.ac.be

Franciska de Jong
Human Media Interaction, Universiteit Twente
Postbus 217, NL-7500 AE Enschede, The Netherlands
f.m.g.dejong@utwente.nl

Charlotte Gooskens
Scandinavisch Instituut, Rijksuniversiteit Groningen
Postbus 716, NL-9700 AS Groningen, The Netherlands
c.s.gooskens@rug.nl

Willemijn Heeren
Human Media Interaction, Universiteit Twente
Postbus 217, NL-7500 AE Enschede, The Netherlands
w.f.l.heeren@ewi.utwente.nl

Feikje Hielkema
Department of Computing Science, University of Aberdeen
Aberdeen AB24 3UE, Scotland, United Kingdom
fhielkem@csd.abdn.ac.uk

Erhard Hinrichs
Seminar für Sprachwissenschaft, Universität Tübingen
Wilhelmstraße 19, D-72074 Tübingen, Germany
eh@sfs.uni-tuebingen.de

Katja Hofmann
ISLA, Informatics Institute, Universiteit van Amsterdam
Kruislaan 403, NL-1098 SJ Amsterdam, The Netherlands
khofmann@science.uva.nl

Dietrick Klakow
Lehrstuhl für Sprachsignalverarbeitung, Universität des Saarlandes
D-66041 Saarbrücken, Germany
dietrick.klakow@lsv.uni-saarland.de

Andreas Merkel
Lehrstuhl für Sprachsignalverarbeitung, Universität des Saarlandes
D-66041 Saarbrücken, Germany
andreas.merkel@lsv.uni-saarland.de

Jens Moberg
Center for Language and Cognition, Rijksuniversiteit Groningen
Postbus 716, NL-9700 AS Groningen, The Netherlands

Paola Monachesi
UiL-OTS, Universiteit Utrecht
Trans 10, NL-3512 JK Utrecht, The Netherlands
paola.monachesi@let.uu.nl

John Nerbonne
Center for Language and Cognition, Rijksuniversiteit Groningen
Postbus 716, NL-9700 AS Groningen, The Netherlands
j.nerbonne@rug.nl

Stephan Oepen
Institutt for informatikk, Universitetet i Oslo
Postboks 1080 Blindern, N-0316 Oslo, Norway
oe@ifi.uio.no

Nelleke Oostdijk
Afdeling Taalwetenschap, Radboud Universiteit Nijmegen
Postbus 9103, NL-6500 HD Nijmegen, The Netherlands
n.oostdijk@let.ru.nl

Roeland Ordelman
Human Media Interaction, Universiteit Twente
Postbus 217, NL-7500 AE Enschede, The Netherlands
ordelman@ewi.utwente.nl

Ineke Schuurman
Centrum voor Computerlinguïstiek, Katholieke Universiteit Leuven
Maria Theresiastraat 21, B-3000 Leuven, Belgium
ineke.schuurman@ccl.kuleuven.be

Nanda Slabbers
Human Media Interaction, Universiteit Twente
Postbus 217, NL-7500 AE Enschede, The Netherlands
slabbers@ewi.utwente.nl

Marco René Spruit
Meertens Instituut
Postbus 94264, NL-1090 GG Amsterdam, The Netherlands
marco.rene.spruit@meertens.knaw.nl

Gerwert Stevens
UiL-OTS, Universiteit Utrecht
Trans 10, NL-3512 JK Utrecht, The Netherlands
gerwert.stevens@let.uu.nl

Daphne Theijssen
Afdeling Taalwetenschap, Radboud Universiteit Nijmegen
Postbus 9103, NL-6500 HD Nijmegen, The Netherlands
daphnetheijssen@student.ru.nl

Mariët Theune
Human Media Interaction, Universiteit Twente
Postbus 217, NL-7500 AE Enschede, The Netherlands
m.theune@ewi.utwente.nl

Jörg Tiedemann
Alfa-Informatica, Rijksuniversiteit Groningen
Postbus 716, NL-9700 AS Groningen, The Netherlands
tiedeman@let.rug.nl

Erik Tjong Kim Sang
ISLA, Informatics Institute, Universiteit van Amsterdam
Kruislaan 403, NL-1098 SJ Amsterdam, The Netherlands
erikt@science.uva.nl

Nathan Vaillette
Department of Mathematics and Computer Science, Dickinson College
Carlisle, PA 17013, Pennsylvania, United States

Tim Van de Cruys
Center for Language and Cognition, Rijksuniversiteit Groningen
Postbus 716, NL-9700 AS Groningen, The Netherlands
t.van.de.cruys@rug.nl

Antal van den Bosch
ILK Research Group, Universiteit van Tilburg
Postbus 90153, NL-5000 LE Tilburg, The Netherlands
antal.vdnbosch@uvt.nl

Laurens van der Werff
Human Media Interaction, Universiteit Twente
Postbus 217, NL-7500 AE Enschede, The Netherlands
laurensw@ewi.utwente.nl

Suzan Verberne
Afdeling Taalwetenschap, Radboud Universiteit Nijmegen
Postbus 9103, NL-6500 HD Nijmegen, The Netherlands
s.verberne@let.ru.nl

Begoña Villada Moirón
Center for Language and Cognition, Rijksuniversiteit Groningen
Postbus 716, NL-9700 AS Groningen, The Netherlands
m.b.villada@let.rug.nl

Eline Westerhout
UiL-OTS, Universiteit Utrecht
Trans 10, NL-3512 JK Utrecht, The Netherlands
eline.westerhout@let.uu.nl

Thomas Zastrow
Seminar für Sprachwissenschaft, Universität Tübingen
Wilhelmstraße 19, D-72074 Tübingen, Germany
post@thomas-zastrow.de