

**10th Conference
on Natural Language Processing (KONVENS)**

**Semantic Approaches
in Natural Language Processing**

**Proceedings of the Conference
on Natural Language Processing 2010**

Edited by
Manfred Pinkal
Ines Rehbein
Sabine Schulte im Walde
Angelika Storrer



universaar
Universitätsverlag des Saarlandes
Saarland University Press
Presses universitaires de la Sarre

Manfred Pinkal, Ines Rehbein, Sabine Schulte im Walde,
Angelika Storrer (Eds.)

**Semantic Approaches
in Natural Language Processing**
**Proceedings of the Conference
on Natural Language Processing 2010**



universaar
Universitätsverlag des Saarlandes
Saarland University Press
Presses universitaires de la Sarre

© September 2010 *universaar*
Universitätsverlag des Saarlandes
Saarland University Press
Presses universitaires de la Sarre



Postfach 151150, 66041 Saarbrücken

ISBN 978-3-86223-004-4 gedruckte Ausgabe
ISBN 978-3-86223-005-1 Online-Ausgabe
URN urn:nbn:de:bsz:291-universaar-124

Projektbetreuung *universaar*: Isolde Teufel

Gestaltung Umschlagseiten: Andreas Franz

Gedruckt auf säurefreiem Papier von Monsenstein & Vannerdat

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <<http://dnb.d-nb.de>> abrufbar.

Contents

Preface	5
Long Papers	
Querying Linguistic Corpora with Prolog <i>Gerlof Bouma</i>	9
Rapid Bootstrapping of Word Sense Disambiguation Resources for German <i>Samuel Broscheit, Anette Frank, Dominic Jehle, Simone Paolo Ponzetto, Danny Rehl, Anja Summa, Klaus Suttnar and Saskia Vola</i>	19
Presupposition Projection and Accommodation in Mathematical Texts <i>Marcos Cramer, Daniel Kuehlwein and Bernhard Schröder</i>	29
Inkrementelle Koreferenzanalyse für das Deutsche <i>Manfred Klenner, Angela Fahrni and Don Tuggener</i>	37
Determining the Degree of Compositionality of German Particle Verbs by Clustering Approaches <i>Natalie Kühner and Sabine Schulte im Walde</i>	47
Noun Phrase Chunking and Categorization for Authoring Aids <i>Cerstin Mahlow and Michael Piotrowski</i>	57
Learning Meanings of Words and Constructions, Grounded in a Virtual Game <i>Hilke Reckman, Jeff Orkin and Deb Roy</i>	67
EXCERPT – Ein integriertes System zum informativen Summarizing und Within-Document-Retrieval <i>Jürgen Reischer</i>	77
Using tf-idf-related Measures for Determining the Anaphoricity of Noun Phrases <i>Julia Ritz</i>	85
Natural Language Processing for the Swiss German Dialect Area <i>Yves Scherrer and Owen Rambow</i>	93
Short Papers	
Clustering Urdu Verbs on the Basis of Related Aspectual Auxiliaries and Light Verbs <i>Tafseer Ahmed</i>	105
Qualitative and Quantitative Error Analysis in Context <i>Christian Chiarcos and Julia Ritz</i>	111
POS-Tagging of Historical Language Data: First Experiments <i>Stefanie Dipper</i>	117
Trainable Tree Distance and an Application to Question Categorisation <i>Martin Emms</i>	123
Training and Evaluating a German Named Entity Recognizer with Semantic Generalization <i>Manaal Faruqui and Sebastian Padó</i>	129
Aufbau eines linguistischen Korpus aus den Daten der englischen Wikipedia <i>Markus Fuchs</i>	135

Detecting Vagueness in Two Languages and Two Domains <i>Viola Ganter</i>	141
Transliteration among Indian Languages using WX Notation <i>Rohit Gupta, Pukit Goyal and Sapan Diwakar</i>	147
Konstruktionsgrammatische Analyse von Gliederungsstrukturen <i>Harald Lünen and Mariana Hebborn</i>	151
Productivity of NPN Sequences in German, English, French, and Spanish <i>Claudia Roch, Katja Keßelmeier and Antje Müller</i>	157
Automatically Determining the Semantic Gradation of German Adjectives <i>Peter Schulam and Christiane Fellbaum</i>	163
Analytic and Synthetic Verb Forms in Irish – An Agreement-Based Implementation in LFG <i>Sebastian Sulger</i>	169
A Base-form Lexicon of Content Words for Correct Word Segmentation and Syntactic-Semantic Annotation <i>Carsten Weber and Johannes Handl</i>	175
Starting a Sentence in L2 German – Discourse Annotation of a Learner Corpus <i>Heike Zinsmeister and Margit Breckle</i>	181

Preface

This proceedings volume contains all long and short papers presented at the 10th biennial conference on Natural Language Processing KONVENTS (Konferenz zur Verarbeitung natürlicher Sprache), which took place on September 6-8, 2010 at Saarland University.

The papers represent a cross-section of recent research in the area of Natural Language Processing. They cover a variety of topics, ranging from NLP applications such as Named Entity Recognition, Word Sense Disambiguation or Part-of-Speech tagging of historical corpora to real-world applications like authoring aids or text summarisation and retrieval. Due to the central theme of the 10th KONVENTS, *Semantic Approaches in Natural Language Processing*, a main focus of the contributions is on linguistic aspects of meaning, covering both deep and shallow approaches to semantic processing, and foundational aspects as well as applications.

The KONVENTS is organised by the scientific societies DGfS-CL (German Linguistic Society, Special Interest Group “Computational Linguistics”), the GSCL (German Society for Computational Linguistics and Language Technology) and the ÖGAI (Austrian Society for Artificial Intelligence).

We would like to thank the Cluster of Excellence “Multimodal Computing and Interaction” at Saarland University as well as the DGFS-CL, whose funding made the conference possible. We would also like to thank the members of the program committee as well as the members of the review board for providing detailed and fair reviews in due time.

These proceedings are also available electronically from Saarland University Press (*universaar*) at

<http://www.uni-saarland.de/de/campus/service-und-kultur/universaar/monographien.html>.

Saarbrücken, September 2010
The editors

Program Committee:

Stephan Busemann
Brigitte Krenn
Manfred Pinkal
Ines Rehbein
Arndt Riester
Sabine Schulte im Walde
Caroline Sporleder
Manfred Stede
Angelika Storrer
Magdalena Wolska

Review Board:

Stephan Bangerter	Brigitte Krenn	Olga Pustylnikov
Ernst Buchberger	Marco Kuhlmann	Michaela Regneri
Stephan Busemann	Lothar Lemnitzer	Arndt Riester
Stefanie Dipper	Harald Lüngen	Josef Ruppenhofer
Kurt Eberle	Torsten Marek	Bernhard Schröder
Christiane Fellbaum	Johannes Matiasek	Caroline Sporleder
Antske Fokkens	Alexander Mehler	Manfred Stede
Anette Frank	Friedrich Neubarth	Stefan Thater
Hagen Fürstenau	Rainer Osswald	Harald Trost
Alexander Geyken	Sebastian Padó	Christina Unger
Jeremy Jancsary	Ulrike Padó	Martin Volk
Bryan Jurish	Alexis Palmer	Kay-Michael Würzner
Alexander Koller	Hannes Pirker	Heike Zinsmeister

LONG PAPERS

Querying Linguistic Corpora with Prolog

Gerlof Bouma

Department Linguistik

Universität Potsdam

Postsdam, Germany

gerlof.bouma@uni-potsdam.de

Abstract

In this paper we demonstrate how Prolog can be used to query linguistically annotated corpora, combining the ease of dedicated declarative query languages and the flexibility of general-purpose languages. On the basis of a Prolog representation of the German Tüba-D/Z Treebank, we show how one can tally arbitrary features of (groups) of nodes, define queries that combine information from different layers of annotation and cross sentence boundaries, query ‘virtual annotation’ by transforming annotation on-the-fly, and perform data driven error analysis. Almost all code needed for these case studies is contained in the paper.

1 Introduction

In recent years, there has been a strong increase in the availability of richly annotated corpora and corpora of ever growing size. In tact with this, there is a thriving research into ways of exploiting these corpora, where especially the conflicting constraints posed by the desire for an expressive query formalism and the computational demands of querying a large corpus form a driving tension. In this paper we hope to contribute to this debate by advocating the use of Prolog, a general-purpose language, to query corpora. We will argue by means of a series of concrete examples that Prolog is declarative enough to allow formulation of corpus queries in an intuitive way, that it is flexible and powerful enough to not constrain the computational linguist wishing to get as much as possible out of a corpus, and that on modern Prolog implementations, it is fast enough to intensively use corpora of a million tokens or more.

Before we turn to the examples that make up the body of this paper, we briefly discuss what makes

Prolog a good language for corpus querying, but also what its disadvantages are. It should be clear from the outset, however, that we do not propose Prolog per se to be used as a query tool for the (non-programmer) general linguist who wants a fully declarative corpus environment, including tree visualization, etc. Rather, our target is the advanced corpus user/computational linguist who needs an extendable query language and features beyond what any specific dedicated query tool can offer, that is, the type of user who will end up using a general-purpose language for part of their corpus tasks.

1.1 Why use Prolog?

Semi-declarativeness, non-deterministic search

Prolog is well-suited to write database-like queries in (see e.g., Nilsson and Maluszynski (1998) for a description of the relation between relational database algebra and Prolog) – one defines relations between entities in terms of logical combinations of properties of these entities. The Prolog execution model is then responsible for the search for entities that satisfy these properties. This is one of the main advantages over other general-purpose languages: in Prolog, the programmer is relieved of the burden of writing functions to search through the corpus or interface with a database.

Queries as annotation Any query that is more complicated than just requesting an entry from the database is a combination of the user’s knowledge of the type of information encoded in the database and its relation to the linguistic phenomenon that the user is interested in. Thus, a query can be understood as adding annotation to a corpus. In Prolog, a query takes the form of a predicate, which can then be used in further predicate definitions. In effect,

we can query annotation that we have ourselves added.

The lack of a real distinction between existing annotation (the corpus) and derived annotation (subqueries) is made even more clear if we consider the possibility to record facts into the Prolog database for semi-permanent storage, or to write out facts to files that can then later be loaded as given annotation. This also opens up the possibility of performing corpus transformations by means of querying. Related to the queries as annotation perspective is the fact that by using a general-purpose programming language, we are not bound by the predefined relations of a particular query language. New relations can be defined, for instance, relations that combine two annotation layers (see also Witt (2005)), or cross sentence boundaries.

Constraining vs inspecting TIGERSearch (König et al., 2003) offers facilities to give (statistical) summaries of retrieved corpus data. For instance, one can get a frequency list over the POS tags of retrieved elements. This is a very useful feature, as it is often such summaries that are relevant. The reversibility of (well-written) Prolog predicates facilitates implementing such functionality. It is possible to use the exact same relations that one uses to constrain query matches to request information about a node. If `has_pos/2` holds between a lexical node and its POS-tag, we can use it to require that a lexical node in a query has a certain POS-tag or to ask about a given node what its POS-tag is.

Scope of quantifiers and negation Query languages differ in the amount of control over the scope of quantifiers and negation in a query (Lai and Bird, 2004). For instance, Kepser's (2003) first-order-logic based query language allows full control over scoping by explicit quantification. On the other hand, TIGERSearch's query language (König et al., 2003) is restrictive as it implicitly binds nodes in a query with a wide scope existential quantifier. Queries like *find an NP that does not contain a Det node* are not expressible in this language.

In a general-purpose language we get full control over scope. We can illustrate this with negation, canonically implemented in Prolog by negation as (Prolog) failure (written: `\+.`). By doing lookup in the database of subtrees/nodes inside or outside of a negated goal, we vary quantifier scope: `lookup(X,`

`\+ p(X)` succeeds when X does not have property p , and `\+ (lookup(X), p(X))` succeeds when there is no X with p . A discussion of the implementation of queries that rely on this precise control over scope can be found in (Bouma, 2010).

1.2 Why not use Prolog?

Not so declarative Compared to dedicated query languages, Prolog lacks declarativeness. The order in which properties are listed in a query may have consequences for the speed with which answers are returned or even the termination of a query. The use of Prolog negation makes this issue even worse. For many queries, there is a simple pattern that avoids the most common problems, though: 1) supply positive information about nodes, then 2) access the database to find suitable candidates, and 3) check negative information and properties that involve arithmetic operations. Most of the examples that we give in the next section follow this pattern.

Poor regular expression support Although there are some external resources available, there is no standardized regular expression support in Prolog. This contrasts with both dedicated query languages and with other high-level general-purpose programming languages. However, for some uses of regular expressions, there are good alternatives. For instance, restricting the POS-tag of a node to a known and finite set of POS-tags could also be achieved through a disjunction or by checking whether the POS-tag occurs in a list of allowed POS-tags. These alternatives are typically easy and fast in Prolog.

After these abstract considerations, we shall spend the rest of the paper looking more concretely at Prolog as a query language in a number of cases. After that, in Section 4, we briefly discuss the speed and scalability of a Prolog-based approach.

2 Exploiting the TüBa-D/Z corpus

In this section, we will demonstrate the flexibility of our approach in a series of small case studies on the TüBa-D/Z treebank of German newspaper articles (Telljohann et al., 2006, v5). The treebank has a size of approximately 800k tokens in 45k sentences, and contains annotation for syntax (topological fields, grammatical functions, phrases, secondary relations) and anaphora. We chose this treebank to be able to show a combination of different annotation layers in our queries.

The section is rather code heavy for a conference paper. By including the lion's share of the Prolog code needed to do the tasks in this section, we intend to demonstrate how concise and quick to set up corpus programming in Prolog can be.

2.1 Corpus representation

We take the primary syntactic trees as the basis of the annotation. Following Brants (1997), we store the corpus as collection of directed acyclic graphs, with edges directed towards the roots of the syntactic trees. A tree from the TüBa-D/Z corpus is represented as a collection of facts `node/7`, which contain for each node: an identifier (a sentence id and a node id), the id of the mother node, the edge label, its surface form, its category or POS-tag and a list with possible other information. Below we see two facts for illustration – a lexical (terminal) node and a phrasal node that dominates it.¹ Phrases carry a dummy surface form '\$phrase'. Secondary edges are represented as `secondary/4` facts, and use the sentence and node ids of the primary trees.

```
% node/7 SentId NodeId MotherId
%                                         Form Edge Cat Other
node(153, 4, 503, die, -, art, [morph=asf]). 
node(153, 503, 508, '$phrase', hd, nx, []). 
% secondary/4 SentId NodeId MotherId Edge
secondary(153, 503, 512, refint).
```

By using the sentence number as the first argument of `node/7` facts, we leverage first argument indexing to gain fast access to any node in the treebank. If we know a node's sentence number, we never need to search longer than the largest tree in the corpus. Since syntactic relations hold within a sentence, querying syntactic structure is generally fast (Section 4). A tree and its full representation is given in Figure 1. We will not use the secondary edges in this paper: their use does not differ much from querying primary trees and the anaphora annotation (Section 2.4). We can define interface relations on these facts that restrict variables without looking up any nodes in the database by partially instantiating them.

```
has_sentid(node(A_s,_,_,_,_,_,_),A_s).
has_nodeid(node(_,A_n,_,_,_,_,_),A_n).
has_mother(node(_,_,A_m,_,_,_,_),A_m).
has_form(node(_,_,_,A_f,_,_,_),A_f).
has_edge(node(_,_,_,A_e,_,_),A_e).
```

¹Coding conventions: Most predicate names are VS(O) sentences: `has_edge(A, A_e)` reads *node A has edge label A_e*. Predicates defined for their side-effects get imperative verbs forms. Variables A,B,C refer to nodes, and subscripts are used for properties of these nodes. Variables that represent updates are numbered A, A₁, A₂. Lists receive a plural-s. For readability, we use the if-then-else notation instead of cuts as much as possible.

```
has_poscat(node(_,_,_,_,_,_),A_p,_).
is_under(A,B):-
    has_mother(A,A_m,A_s),
    is_phrasal(B),
    has_nodeid(B,A_m,A_s).

is_under_as(A,B,A_e):-
    is_under(A,B),
    has_edge(A,A_e).

are_sentmates(A,B):-
    has_sentid(A,A_s),
    has_sentid(B,A_s).

is_phrasal(A):-
    has_form(A,'$phrase').
```

Actually looking up a node in the corpus involves calling a term `node/7`, for instance by defining a property of a node-representing variable and then calling the variable: `is_phrasal(A)`, *A* will succeed once for each phrasal node in the corpus.

Transitive closures over the simple relations above define familiar predicates such as dominance (closure of `is_above/2`). In contrast with the simple relations, these closures do look up (instantiate) their arguments. In addition, `has_ancestor/3` also returns a list of intermediate nodes.

```
has_ancestor(A,B):-
    has_ancestor(A,B,_).

has_ancestor(A,B,AB_path):-
    are_sentmates(A,B),
    A, is_under(A,A1), A1,
    has_ancestor_rfl(A1,B,AB_path).
    has_ancestor_rfl(A,A,[ ]).

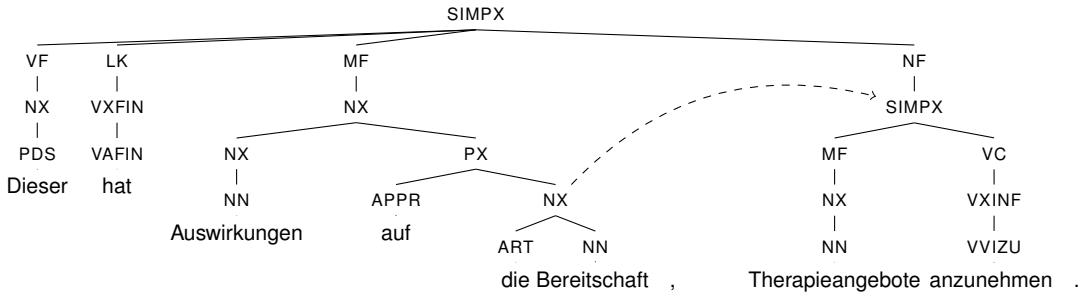
has_ancestor_rfl(A,B,[A|AB_path]):-
    is_under(A,A1), A1,
    has_ancestor_rfl(A1,B,AB_path).
```

With these basic relations in place, let us look at determining linear order of phrases.

2.2 Linear order

As yet, linear order is a property of words in the string (lexical nodes), that we may determine by looking at their node ids (cf. Figure 1). Linear order of phrases is not defined. We can define the position of *any* node as its span over the string, which is determined by the outermost members in the node's yield, that is, the members of the yield with the minimum and maximum node id.²

²`fold/3` (left fold, aka *reduce*) and `map/N` are higher order predicates that generalize predicates to apply to list(s) of arguments, familiar from functional programming (see e.g., Naish (1996)). Given that `min/3` relates two numbers and their minimum, the goal `fold(min,Ns,M)` succeeds if M is the lowest number in of the list Ns. Given that `has_nodeid/2` relates one node and its id, the goal `map(has_nodeid,As,A_ns)` succeeds on a list of nodes As and a list of their corresponding ids A_ns.



"This has effects on the willingness to accept therapy."

```

node(153, 0, 500, 'Dieser', hd, pds, [morph=nsml]).  

node(153, 1, 501, hat, hd, vafin, [morph='3sis']).  

node(153, 2, 502, 'Auswirkungen', hd, nn, [morph=apf]).  

node(153, 3, 508, auf, -, appr, [morph=a]).  

node(153, 4, 503, die, -, art, [morph=asf]).  

node(153, 5, 503, 'Bereitschaft', hd, nn, [morph=asf]).  

node(153, 6, 0, '(', --, '$,, [morph= --]).  

node(153, 7, 504, 'Therapieangebote', hd, nn, [morph=apn]).  

node(153, 8, 505, anzunehmen, hd, vvizu, [morph= --]).  

node(153, 9, 0, '.', --, $., [morph= --]).  
  

secondary(153,503,512,refint).

```

```

node(153, 515, 0, '$phrase', --, simpx, []).  

node(153, 506, 515, '$phrase', -, vf, []).  

node(153, 500, 506, '$phrase', on, nx, []).  

node(153, 507, 515, '$phrase', -, lk, []).  

node(153, 501, 507, '$phrase', hd, vxfin, []).  

node(153, 513, 515, '$phrase', -, mf, []).  

node(153, 511, 513, '$phrase', oa, nx, []).  

node(153, 502, 511, '$phrase', hd, nx, []).  

node(153, 508, 511, '$phrase', -, px, []).  

node(153, 503, 508, '$phrase', hd, nx, []).  

node(153, 514, 515, '$phrase', -, nf, []).  

node(153, 512, 514, '$phrase', mod, simpx, []).  

node(153, 509, 512, '$phrase', -, mf, []).  

node(153, 504, 509, '$phrase', oa, nx, []).  

node(153, 510, 512, '$phrase', -, vc, []).  

node(153, 505, 510, '$phrase', hd, vxinf, []).

```

Figure 1: A tree from Tüba-D/Z and its Prolog representation.

```

spans(A,A_beg,A_end):-  

    yields_dl(A,Bs\[]),  

    map(has_nodeid,Bs,B_ns),  

    fold(min,B_ns,A_beg),  

    fold(max,B_ns,B_n_mx),  

    A_end is B_n_mx+1

```

The yield of a phrase is the combined yields of its daughters. A lexical node is its own yield.

```

yields_dl(A,Bs):-  

    is_phrasal(A)  

    -> ( is_above(A,A1),  

        findall(A1, A1, A1s),  

        map(yields_dl,A1s,Bss),  

        fold	append_dl,Bss,Bs)  

    )  

; % is_lexical(A)  

Bs = [A|Cs]\Cs.

```

According to this definition, the span of the word Auswirkungen in the tree in Figure 1 is 2–3, and the span of the MF-phrase is 2–6.

It makes sense to store the results from spans/2 instead of recalculating them each time, especially if we intend to use this information often. Using a node's span, we can define other relations, such as precedence between phrasal and lexical nodes, and edge alignment.

```

precedes(A,B):-  

    are_sentmates(A,B),  

    spans(A,_,A_end),  

    spans(B,B_beg,_),  

    A_end =< B_beg.  
  

are_right_aligned(A,B):-  

    are_sentmates(A,B),

```

```

spans(A,_,A_end),  

spans(B,_,A_end).

```

Although there are no discontinuous primary phrases in Tüba-D/Z, the definition of precedence above would be appropriate for such phrases, too. Note, however, that in this case two nodes may be unordered even when one is not a descendant of the other. In TIGERSearch, precedence between phrases is defined on both left corners (König et al., 2003). It would be trivial to implement this alternative.

2.3 Phrase restricted bigrammes

In the first task, we see a combination of negation scoping over an existential quantifier in a query and the use of a predicate to ask for a property (surface form) rather than to constrain it. The task is to retrieve token bigrammes contained in non-recursive NPs, which are NPs that do not contain other NPs. This requires a negation scoping over the selection of a descendent NP node. Once we have a non-recursive NP, we select adjacent pairs of nodes from its ordered yield and return the surface forms:

```

bigr_in_nonrec_NP(A_f,B_f):-  

    has_form(A,A_f),  

    has_form(B,B_f),  

    has_cat(C,nx),  

    has_cat(D,nx),  

    C, \+ has_ancestor(D,C),  

    yields_ord(C,Es),  

    nextto(A,B,Es).

```

`yields_ord/2` holds between a node and its ordered yield. Ordering is done by decorate-sort-undecorate.

```
yields_ord(A,Bs) :-
    yields_dl(A,Cs\[]),
    keys_values(CnsCs,C_ns,Cs), % decorate
    map(has_nodeid,Cs,C_ns),
    keysort(CnsCs,BnsBs),      % sort
    values(BnsBs,Bs).          % undecorate
```

The query succeeds 160968 times, that is, there are 161k bigramme tokens in non-recursive NPs in the corpus. There are 105685 bigramme types, of which the top 10 types and frequencies are:

(1)	1	der Stadt	141	6	der Welt	103
	2	der Nato	138	7	den letzten	103
	3	mehr als	136	8	die Polizei	98
	4	die Nato	125	9	ein paar	97
	5	die beiden	107	10	den USA	90

2.4 Combining annotation layers

As mentioned, the Tüba-D/Z corpus additionally contains annotation of anaphora and coreference. This annotation layer can be considered as a graph, too, and may be stored in a fashion similar to the secondary edges. The sentence and node ids in the `anaphor/5` facts are again based on the primary trees.

```
% anaphor/4 SentId NodeId Rel SentIdM NodeIdM
anaphor(4, 527, coreferential, 1, 504).
anaphor(4, 6, anaphoric, 4, 527).
anaphor(6, 522, coreferential, 4, 512).
```

In addition, we have a convenience predicate that links `node/7` terms to the anaphora facts.

```
is_linked_with(A,Rel,B) :-
    has_nodeid(A,A_n,A_s),
    has_nodeid(B,B_n,B_s),
    anaphor(A_s,A_n,Rel,B_s,B_n).
```

A very basic combination of the two annotation layers allows us to formulate the classic *i*-within-*i* constraint (Hoeksema and Napoli, 1990).

```
% [ ... X_i ... ]_i
i_within_i(A,B) :-
    is_linked(A,_Rel,B),
    has_ancestor(A,B).
```

The query returns 19 hits, amongst which (2):

- (2) [die kleine Stadt mit ihren, 7.000 Einwohnern],
the small town with its inhabitants

2.5 Grammatical function parallelism

In anaphora annotation, links can be made between nodes that are not contained within one sentence – a coreference link could span the entire corpus. In this task, we follow intra-sentential links. We will try to find corpus support for the (not uncontested) claim that people prefer to interpret pronouns such

Pron. GF	Antecedent GF				Total
	on	oa	od	rest	
on	2411 .03	236 .09	162 .06	589	3398
oa	168 .18	46 .73	16 .08	62	292
od	173 .03	20 .02	19 .38	45	257
rest	142	17	15	63	237
Total	2894	319	212	759	4184

Table 1: Cross tabulation of grammatical function of pronoun-antecedent pairs from adjacent sentences, counts and association scores (PMI).

that the antecedent and pronoun have the same grammatical function (Smyth, 1994, a.o.). As a reflection of this preference, we might expect that there is a trend for a pronoun and its antecedent in the directly preceding sentence to have the same grammatical function. The predicate `ana_ant_gf/2`, defined below, returns the grammatical functions of an anaphoric NP headed by a personal pronoun and its NP antecedent if it occurs in the immediately preceding sentence.

```
ana_ant_gf(A_e,B_e) :-
    has_edge(A,A_e,A_s),
    is_under_as(A1,A,hd),
    has_pos(A1,pper),
    has_edge(B,B_e,B_s),
    has_cat(B,nx),
    is_linked_with(A,anaphoric,B),
    B_s is A_s-1, % B in sentence before A
    A, A1, B.
```

The query succeeds just over 4k times. Table 1 summarizes the results. For the top-left cells, we've calculated pointwise association (PMI) between the two variables. The rows on the diagonal have positive values, which means the combination occurs more often than expected by chance, as expected by the parallelism hypothesis.³ In Section 2.7, we will revisit this task.

2.6 Coreference chains

Until now, we have used the anaphoric annotation as-is. However, we can also consider it in terms of coreference chains, rather than single links between nodes. That is, we can construct equivalence classes of nodes that are (transitively) linked to each other: they share one discourse referent. Naïve

³This should not be taken as serious support for the hypothesis, though. The association values are small and there are other positive associations in the table. Also, we have not put a lot of thought into how the annotation relates to the linguistic phenomenon that we are trying to investigate.

construction of such classes is hampered by the occurrence of cycles in the anaphora graph. Therefore, we need to check for each anaphoric node whether its containing graph contains a cycle. If it does, we pick any node in the cycle as the root of the graph. Non-cyclic graphs have the (unique) node with out-degree zero as their root. We use the Prolog database to record the roots of cyclic graphs, so that we can pick them as the root next time we come to this graph.

```

has_root(A,B) :-
    is_linked_with(A,A1),
    ( leads_to_cycle_at(A1,C)
    -> ( B = C,
          record_root(B)
        )
    ; has_root_rfl_nocycles(A,B)
    ).

has_root_rfl_nocycles(A,B) :-
    is_recorded_root(A)
    -> B = A
    ; is_linked_with(A,A1)
    -> has_root_rfl_nocycles(A1,B)
    ; B = A.

```

The cycle check itself is based on Floyd's tortoise and hare algorithm, whose principle is that if a slow tortoise and a fast hare traversing a graph land on the same node at the same time, there has to be a cycle in the graph. In our version, the tortoise does not traverse already recorded root nodes, to prevent the same cycle from being detected twice.

```

leads_to_cycle_at(A,B) :-
    is_linked_with(A,A1),
    is_linked_with(A1,A2),
    tortoise_hare(A1,A2,B).

tortoise_hare(A,B,C) :-
    \+ is_recorded_root(A),
    ( A = B % evidence of cycle
    -> C = A
    ; ( is_linked_with(A,A1), % tort. to A1
        is_linked_with(B,B1), % hare to
        is_linked_with(B1,B2), % B2
        tortoise_hare(A1,B2,C)
      )
    ).

```

Collecting all solutions for `has_root/2`, we find 20516 root and 50820 non-root referential expressions. The average coreference chain length is 3.48, the longest chain has 176 mentions in 164 sentences.

2.7 Revisiting parallelism

Let us go back to pronoun-antecedent parallelism with this alternative view of the anaphora annotation. In our first attempt, we missed cases where a pronoun's referent is mentioned in the previous sentence, just not in a node directly anaphorically linked to the pronoun. The coreference chain

Pron. GF	Antecedent GF				Total
	on	oa	od	rest	
on	4563 .02	500 -.07	353 -.05	1312	6728
oa	367 -.13	85 .53	43 .21	134	629
od	392 -.02	48 .00	47 .34	115	602
rest	309	39	26	136	510
Total	5631	672	469	1697	8469

Table 2: Cross tabulation of grammatical function of pronoun-antecedent pairs from adjacent sentences, counts and association scores (PMI). Revisited.

view gives us a chance to get at these cases. Note that now a pronoun may have more than one antecedent in the preceding sentence. The predicate `ana_ant_gf_rev/2` succeeds once for each of the possible pairs:

```

ana_ant_gf_rev(A_e,B_e) :-
    has_edge(A,A_e,A_s),
    is_under_as(A1,A,hd),
    has_pos(A1,ppr),
    has_edge(B,B_e,B_s),
    has_cat(B,nx),
    A1, A,
    B_s is A_s-1,
    corefer(A,B).

```

Coreference between two given and distinct nodes can be defined on `has_root/2`. That is, the definition of coreference relies on a transformation of the original annotation, that we are producing on-the-fly.

```

corefer(A,B) :-
    has_root(A,B).
corefer(A,B) :-
    has_root(B,A).
corefer(A,B) :-
    has_root(A,C),
    has_root(B,C).

```

Just like in our first attempt, we collect the results in a table and calculate observed vs expected counts. As could be expected, we get many more datapoints ($\sim 8.5k$). Table 2 shows a similar picture as before: small, positive associations in the diagonal cells.

3 Corpus inspection

With the techniques introduced thus far, we can perform corpus inspection by formulating and calling queries that violate corpus well-formedness constraints. A data-driven, large scale approach to error mining is proposed by Dickinson and Meurers (2003). Errors are located by comparing the analyses assigned to multiple occurrences of the same

string. A version of this idea can be implemented in the space of this paper. Naïve comparison of all pairs of nodes would take time quadratic in the size of the corpus. Instead, we record the string yield and category of each interesting phrasal node in the database, and then retrieve conflicts by looking for strings that have more than one analysis. First, an interesting phrase is one that contains two or more words, unary branching supertrees.

```
interesting_node(A,Bs) :-
    phrasal(A), A,
    \+ is_alone_under(_,A),
    Bs = [_,_|_], % >= two nodes in yield
    yields_ord(A,Bs).
```

```
is_alone_under(A,B) :-
    is_under(A,B), A, B,
    \+ ( is_above(B,A1), A1, A1\=A ).
```

Then, recording a string involves checking whether we have already seen it before and, if so, whether we have a new analysis or an existing one.

```
record_string_analysis(A,Bs) :-
    map(has_form,Bs,B_fs),
    fold(spaced_atom_concat,B_fs,String),
    ( retract(str_analyses(String,Analyses))
        -> insert_analysis(A,Analyses,Analyses1)
    ; insert_analysis(A,[],Analyses1)
    ),
    assert(str_analyses(String,Analyses1)).
insert_analysis(A,Analyses,Analyses1) :-
    has_cat(A,A_c),
    ( select(A_c-As,Analyses,Analyses2)
        -> Analyses1 = [A_c-[A|As]|Analyses2]
    ; Analyses1 = [A_c-[A]|Analyses]
    ).
```

Exhaustively running the two main queries asserts 364785 strings into the database, with averages of 1.0005 different categories per string and 1.1869 occurrences per string.

The query `str_analyses(Str,[_,_|_])` succeeds 178 times, once for each string with more than one analysis in the corpus. Far from all of these are true positives. Common false positives are forms that can be AdvPs (ADVX), NPs (NX) or DPs, such as *immer weniger* ‘less and less’:

- (3) das Flügelspiel fand [ADVX
the piano playing found
immer weniger] statt
less and less place
‘The piano was played less and less often.’
- (4) Japan importiert [NX immer weniger]
Japan imports less and less
‘Japan imports fewer and fewer goods.’
- (5) Die braven BürgerInnen produzieren [DP
Those good citizens produce
immer weniger] Müll
less and less waste

‘The good citizens produce less and less waste’

We also see borderline cases of particles that might or might not be attached to their neighbours:

- (6) Für Huhn ungewöhnlich saftig [MF auch sie]
for chicken remarkably juicy also it
‘It, too, was remarkably juicy, for being chicken.’
- (7) Wahrscheinlich streift [NX auch sie] in diesem
Probably roams also she at this
Moment durch ihr Nachkriegsberlin.
moment through her post-war Berlin
‘Probably, she, too, roams through her post-war Berlin at this moment.’

In (6), the node of interest is labelled MF for the topological Mittelfeld. This is not a traditional constituent, but since Tüba-D/Z annotates topological fields we also capture some cases where a string is a constituent in one place and a non-constituent in another. Dickinson and Meurers (2003) introduce dummy constituents to systematically detect a much wider range of those cases.

Finally, real errors include the following example of an NP that should have been an AdjP:

- (8) [NX Drei Tage lang] versuchte Joergensen ...
three days long tried Joergensen
‘Joergensen tried for three days to ...’
- (9) [ADJX Drei Tage lang] versuchten hier
three days long tried here
Museumsarchitekten ...
museum architects
‘Museum architects tried for three days to ...’

The proposal in Dickinson and Meurers (2003) is more elaborate than our implementation here, but it is certainly possible to extend our setup further. We have shown that a basic but flexible query environment is quick to set up in Prolog. Prolog makes a suitable tool for corpus investigation and manipulation because it is a general-purpose programming language that by its very nature excels at programming in terms of relations and non-deterministic search.

4 Performance

With ever growing corpora, speed of query evaluation becomes a relevant issue. To give an idea of the performance of our straightforward use of Prolog, Table 3 shows wall-clock times of selected

Task	# Solutions	Time
Loading & indexing corpus		31s
Loading & indexing compiled corpus		3s
<code>lookup/1</code>	1741889	2s
<code>yields_ord/2</code>	1741889	81s
<code>spans/3</code>	1741889	87s
<code>bigr_in_nonrec_NP/2</code>	160968	80s
<code>i_within_i/2</code>	19	1s
<code>has_root/2</code>	50820	5s
<code>ana_ant_gf/2</code>	4184	1s
<code>ana_ant_gf_rev/2</code>	8471	9s
<code>record_str_analyses/0</code>	1	101s
<code>inconsistency/2</code>	178	1s

Table 3: Wall-clock times of selected tasks.

tasks.⁴

The uncompiled corpus of 45k sentences ($\sim 1.8M$ Prolog facts) loads in about half a minute, but using precompiled prolog code – an option many implementations offer – reduces this to 3 seconds. The bottom of the table gives the time it takes to calculate the number of solutions for queries described in the previous section, plus `lookup/1` which returns once for each node in the corpus. As can be seen, queries are generally fast, except for those that involve calculating the yield. The use of memoization or even pre-computation would speed these queries up. Memory consumption is also moderate: even with `record_str_analyses/0`, the system runs in around 0.5Gbytes of RAM.

As an indication of the scalability of our approach, we note that we (Bouma et al., 2010) have run queries on dependency parsed corpora of around 40M words (thus 40M facts). Loading such a corpus takes about 10 minutes (or under 1 minute when precompiled) and uses 13GByte on a 64bit machine. Because of first-argument indexing on sentence ids, time per answer does not increase noticeably. We conclude that the approach in this paper scales to at least medium-large corpora. Scaling to even larger corpora remains a topic for future investigation. Possible solutions involve connecting Prolog to an external database, or (as a low-tech alternative) sequential loading of parts of the corpus.

5 Conclusions

In this paper, we hope to have shown the merits of Prolog as a language for corpus exploitation with the help of a range of corpus tasks. It is a

⁴Test machine specifications: 1.6Ghz Intel Core 2 Duo, 2GBytes RAM, SWI-prolog v5.6 on 32-bit Ubuntu 8.04

flexible and effective language for corpus programming. The fact that most Prolog code needed for our demonstrations is in this paper makes this point well. Having said that, it is clear that the approach demonstrated in this paper is not a complete replacement of dedicated query environments that target non-programmers. In depth comparison with alternatives – corpus query environments, general-purpose language libraries, etc. – is beyond the scope of this paper, but see Bouma (2010) for a comparison of Prolog’s performance and expressiveness with TIGERSearch on number of canonical queries.

Future work will include the investigation of techniques from constraint-based programming to make formulating queries less dependent on the procedural semantics of Prolog and the exploitation of corpora that cannot be fitted into working memory.

Our studies thus far have resulted not only in queries and primary code, but also in conversion scripts, auxiliary code for pretty printing, etc. We intend to collect all these and make these available on-line, so as to help interested other researchers to use Prolog in corpus investigations and to facilitate reproducibility of studies relying on this code.

References

- Gerlof Bouma, Lilja Øvrelid, and Jonas Kuhn. 2010. Towards a large parallel corpus of cleft constructions. In *Proceedings of LREC 2010*, pages 3585–3592, Malta.
- Gerlof Bouma. 2010. Syntactic tree queries in Prolog. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 212–217, Uppsala.
- Thorsten Brants. 1997. The NEGRA export format. Technical report, Saarland University, SFB378.
- Markus Dickinson and Detmar Meurers. 2003. Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*, Växjö.
- Jack Hoeksema and Donna Jo Napoli. 1990. A condition on circular chains: A restatement of i-within-i. *Journal of Linguistics*, 26(2):403–424.
- Stephan Kepser. 2003. Finite structure query - a tool for querying syntactically annotated corpora. In *Proceedings of EACL 2003*, pages 179–186.
- Esther König, Wolfgang Lezius, and Holger Voermann. 2003. Tigersearch 2.1 user’s manual. Technical report, IMS Stuttgart.

Catherine Lai and Steven Bird. 2004. Querying and updating treebanks: A critical survey and requirements analysis. In *Proceedings of the Australasian Language Technology Workshop*, Sydney.

Lee Naish. 1996. Higher-order logic programming in Prolog. Technical Report 96/2, Department of Computer Science, University of Melbourne, Melbourne, Australia, February.

Ulf Nilsson and Jan Maluszynski. 1998. *Logic, programming and Prolog*. John Wiley & Sons, 2nd edition.

Ron Smyth. 1994. Grammatical determinants of ambiguous pronoun resolution. *Journal of Psycholinguistic Research*, 23:197–229.

Heike Telljohann, Erhard Hinrichs, Sandra Kübler, and Heike Zinsmeister. 2006. Stylebook for the Tübingen treebank of written German (TüBa-D/Z). revised version. Technical report, Seminar für Sprachwissenschaft, Universität Tübingen.

Andreas Witt. 2005. Multiple hierarchies: New aspects of an old solution. In Stefanie Dipper, Michael Götze, and Manfred Stede, editors, *Heterogeneity in Focus: Creating and Using Linguistic Databases*, Interdisciplinary Studies on Information Structure (ISIS) 2, pages 55–86. Universitätsverlag Potsdam, Potsdam.

Rapid Bootstrapping of Word Sense Disambiguation Resources for German

**Samuel Broscheit, Anette Frank, Dominic Jehle, Simone Paolo Ponzetto,
Danny Rehl, Anja Summa, Klaus Suttnar and Saskia Vola**

Department of Computational Linguistics

Heidelberg University

lastname@cl.uni-heidelberg.de

Abstract

This paper presents ongoing efforts on developing Word Sense Disambiguation (WSD) resources for the German language, using GermaNet as a basis. We bootstrap two WSD systems for German. (i) We enrich GermaNet with *predominant sense information*, following previous unsupervised methods to acquire predominant senses of words. The acquired predominant sense information is used as a type-based first sense heuristics for token-level WSD. (ii) As an alternative, we adapt a state-of-the-art *knowledge-based WSD system* to the GermaNet lexical resource. We finally investigate the hypothesis of whether the two systems are complementary by combining their output within a voting architecture. The results show that we are able to bootstrap two robust baseline systems for word sense annotation of German words.

1 Introduction

Word Sense Disambiguation (WSD), the task of computationally determining the meanings of words in context (Agirre and Edmonds, 2006; Navigli, 2009), is a well-studied Natural Language Processing (NLP) task. In comparison to other labeling tasks, WSD is highly challenging because of the large amount of different senses that words have in context, and the difficulty of discriminating them, given the fine-grained sense distinctions offered by existing lexical resources.

If one relies on a fixed sense inventory, two main approaches have been proposed in the literature: supervised and knowledge-based methods. Both methods crucially require language resources in the form of wide-coverage semantic lexica or annotated data. While the most successful approaches to WSD are based on super-

vised machine learning, these require large training sets of sense-tagged data, which are expensive to obtain. Knowledge-based methods minimize the amount of supervision by exploiting graph-based algorithms on structured lexical resources such as WordNet (Fellbaum, 1998). Following the model of WordNet, wordnets have been developed for a wide range of languages (Vossen, 1998; Lemnitzer and Kunze, 2002; Pianta et al., 2002; Atserias et al., 2004; Tuñç et al., 2004, *inter alia*). Moreover, research efforts recently focused on automatically acquiring wide-coverage multilingual lexical resources (de Melo and Weikum, 2009; Mausam et al., 2009; Navigli and Ponzetto, 2010, *inter alia*).

An alternative to supervised and knowledge-based approaches is provided by fully unsupervised methods (Schütze, 1992; Schütze, 1998; Pedersen and Bruce, 1997, *inter alia*), also known as word sense induction approaches: these merely require large amounts of raw text, but do not deliver well-defined sense clusters, and therefore are more difficult to exploit.

For supervised WSD methods, corpora such as SemCor (Miller et al., 1993) and the ones developed for the SensEval (Mihalcea and Edmonds, 2004) and SemEval (Agirre et al., 2007) competitions represent widely-used training resources. However, in the case of German, the development of supervised WSD systems based on the sense inventory provided by GermaNet (Lemnitzer and Kunze, 2002) is severely hampered by the lack of annotated corpora.

This lack of a sense-annotated corpus implies in turn that no predominant sense information is available for GermaNet senses, in contrast to WordNet, which offers Most Frequent Sense (MFS) information computed from fre-

quency counts over sense annotations in SemCor. As a result, no MFS baseline system can be produced for German data, and no MFS heuristics, i.e. assigning the predominant sense in case no answer can be computed, is available. To overcome these limitations, we propose to leverage existing proposal for English and exploit them to bootstrap new WSD resources for German. Our contributions are the following:

1. We *enrich GermaNet with predominant sense information* acquired from large web-based corpora, based on previous work on unsupervised predominant sense acquisition for English words. This allows us to automatically label target words in context using the predominant sense as a type-level first-sense heuristics.
2. We adapt a *state-of-the-art knowledge-based WSD system* to tag words in context with GermaNet senses. This system performs an instance-based disambiguation based on contextual information, and allow us to move away from the type-based first-sense heuristics.
3. We explore the hypothesis of whether the word sense annotations generated by these two WSD approaches are complementary, and accordingly experiment with combining their outputs in a *voting architecture*.

The remainder of the paper is structured as follows. Section 2 presents how we adapted previous proposals for finding predominant senses and performing graph-based WSD in English for German. In Section 3 we present a gold standard we created for German WSD and report our experiments and evaluation results. Section 4 concludes the paper.

2 Rapid Bootstrapping of WSD Resources for German

Our approach to develop WSD resources for German is two-fold. We first apply state-of-the-art methods to find predominant senses for English (McCarthy et al., 2004; Lapata and Keller, 2007, Sections 2.1 and 2.2). Both methods are language-independent and require minimal supervision. However, they do not make use of the structured knowledge provided by the GermaNet taxonomy. Accordingly, in Section 2.3 we move on and adapt the state-of-the-art graph-based WSD system of Agirre and Soroa (2009) to use GermaNet as a lexical knowledge base. Finally,

we propose to integrate the output of these methods in Section 2.4.

2.1 Using a Thesaurus-based Method

McCarthy et al. (2004) propose an unsupervised method for acquiring the predominant sense of a word from text corpora. Key to their approach is the observation that distributionally similar words of a given target word tend to be sense-related to the sense of the target word. Thus, for a set of distributionally similar words N_w of a target word w , they compute semantic similarity according to some WordNet similarity measure for each pair of senses of w and senses of n_j , for all $n_j \in N_w$. The WordNet-based semantic similarity scores (sss) are weighted by the distributional similarity scores (dss) of the respective neighbors:

$$\text{prevalence_score}(w, s_i) = \sum_{n_j \in N_w} dss(w, n_j) \times \frac{sss(s_i, n_j)}{\sum_{s'_i \in \text{senses}(w)} sss(s'_i, n_j)} \quad (1)$$

where $sss(s_i, n_j) = \max_{s_x \in \text{senses}(n_j)} sss(s_i, s_x)$

Choosing the highest-scoring sense for w yields the predominant sense tailored to the domain of the underlying corpus on which distributional similarity is computed. McCarthy et al. (2004) make use of Lin’s (1998) method of constructing a thesaurus of distributionally similar words. Such a thesaurus can be computed on the basis of grammatical relations or word proximity relations from parsed or raw text corpora, respectively. Syntactic relations were extracted with RASP (Briscoe et al., 2006) from 90M words of the BNC (Leech, 1992). WordNet-based semantic similarity was computed using the *jcn* (Jiang and Conrath, 1997) and *lesk* (Banerjee and Pedersen, 2003) measures.

The acquired predominant senses were evaluated against SemCor for the different parts of speech, both at the type-level (measuring accuracy of predicting the predominant sense of words within SemCor) and for tokens (measuring the accuracy of using the predominant sense as a first sense heuristics in instance-based sense tagging). For both evaluations, the predominant senses calculated perform well over a random baseline. Compared to the most frequent sense computed from SemCor, the predominant senses score lower. For instance, for nouns using BNC and *lesk* they report 24.7% random baseline, 48.7% predomi-

nant sense and 68.6% MFS accuracy. Verbs, adjectives and adverbs show the same pattern at lower performance levels.

Computing predominant senses for German.

To acquire predominant sense information for German using McCarthy et al.'s (2004) method, we first need a large corpus for the computation of distributional similarity. We select the German part of the WaCky corpora (Baroni et al., 2009), deWAC henceforth, a very large corpus of 1.5G words obtained by web crawling, additionally cleaned and enriched with basic linguistic annotations (PoS and lemma). For parsing we selected a subcorpus of sentences (i) that are restricted to sentence length 12 (to ensure good parsing quality) and (ii) that contain target words from GermaNet version 5.1 (nouns, adjectives and verbs). From this subcorpus we randomly selected sample sets for each word for parsing. Parsing was performed using Malt parser (Hall and Nivre, 2008), trained on the German TüBa/DZ corpus. The parser output is post-processed for special constructions (e.g. prepositional phrases, auxiliary constructions), and filtered to reduce dependency triples to semantically relevant word pairs. The computation of distributional similarity follows Lin (1998), whereas semantic similarity is computed using Leacock & Chodorow's (1998) measure, built as an extension of the GermaNet API of Gurevych and Niederlich (2005).

Predominant sense scores are computed according to Equation 1. To determine optimal settings of system parameters, we made use of a held-out development set of 20 words. We obtained the best results for this set using subcorpora for words with 20-200 occurrences in deWAC, a selection of up to 200 sentences per word for dependency extraction, and restriction to 200 nearest neighbors from the set of distributionally similar words for prevalence score computation.

We developed two components providing predominant sense annotations using GermaNet senses: the computed predominant senses are included as an additional annotation layer in the deWAC corpus. Moreover, we extended the GermaNet API of Gurevych and Niederlich (2005) to return predominant senses, which implements a baseline system for predominant sense annotation.

2.2 Using an Information Retrieval Approach

Lapata and Keller (2007) present an information retrieval-based methodology to compute sense predominance which, in contrast to McCarthy et al. (2004), requires no parsed text. Key to their approach is to query an information retrieval system to estimate the degree of association between a word and its sense descriptions as defined by WordNet synsets. That is, predominant senses are automatically discovered by computing for each sense of a target word how often the word co-occurs with the synonyms of that sense. Let w be a target word and SD_{s_i} the sense description for s_i , namely the i -th sense of w . In practice, SD_{s_i} is a set of words $\{w_1 \dots w_n\}$ which are strongly semantically associated with s_i , e.g. its synonyms, and provide a context for sense ranking. The predominant sense is then obtained by selecting the sense description which has the highest co-occurrence score with w :

$$\hat{s} = \underset{s_i \in \text{senses}(w)}{\operatorname{argmax}} df(\{w\} \cup SD_{s_i})$$

where df is a document frequency score, i.e. the number of documents that contain w and words from SD_{s_i} (which may or may not be adjacent), as returned from queries to a text search engine. The queries are compiled for all combinations of the target word with each of its synonyms, and the frequencies are combined using different strategies (i.e. sum, average or taking the maximum score).

Computing predominant senses for German. We start with a German polysemous noun, e.g. *Grund* and collect its senses from GermaNet:

nнатGegenstand.3	{Land}
nnatGegenstand.15	{Boden, Gewaesser}
nArtefakt.6305	{Boden, Gefaess}
nMotiv.2	{Motivation, Beweggrund, Veranlassung, Anlass}.

We then compile the queries, e.g. in the case of the nMotiv.2 sense the following queries are created

```
Grund AND Motivation
Grund AND Beweggrund
Grund AND Veranlassung
Grund AND Anlass
```

and submitted to a search engine. The returned document frequencies are then normalized by the document frequency obtained by querying the synonym alone. Finally the senses of a word are ranked according to their normalized frequency,

and the one with the highest normalized frequency is taken as the predominant sense.

Lapata and Keller (2007) explore the additional expansion of the sense descriptors with the hypernyms of a given synset. While their results show that models that do not include hypernyms perform better, we were interested in our work in testing whether this holds also for German, as well as exploring different kinds of contexts. Accordingly, we investigated a variety of extended contexts, where the sense descriptions include synonyms together with: (i) paraphrases, which characterize the meaning of the synset (PARA, e.g. *Weg* as a ‘often not fully developed route, which serves for walking and driving’); (ii) hypernyms (HYPER, à la Lapata and Keller (2007)); (iii) hyponyms (HYPO) (iv) all hyponyms together in a disjunctive clause e.g. “Grund AND (Urgrund OR Boden OR Naturschutzgrund OR Meeresgrund OR Meeresboden)” (HYPOALL). The latter expansion technique is motivated by observing during prototyping that one hyponym alone tends to be too specific, thus introducing sparseness. In order to filter out senses which have sparse counts, we developed a set of heuristic filters:

- LOW FREQUENCY DIFFERENCE (FREQ) filters out *words* whose difference between the relative frequencies of their first two synsets falls below a confidence threshold, thus penalizing vague distinctions between senses.
- LOW DENOMINATOR COUNT (DENOM) removes *synsets* whose denominator count is too low, thus penalizing synsets whose information in the training corpus was too sparse.
- LOW SYNSET INFORMATION COUNT (SYN) filters out *synsets* whose number of synonyms falls under a confidence threshold.

In our implementation, we built the information retrieval system using Lucene¹. Similarly to the setting from Section 2.1, the system was used to index the deWAC corpus (Baroni et al., 2009). Due to the productivity of German compounds, e.g. “Zigarettenanzündersteckdose” (cigarette lighter power socket), many words cannot be assigned word senses since no corresponding lexical unit can be found in GermaNet. Accordingly, given a compound, we perform a morphological analysis to index and retrieve its lexical

head. We use Morphisto (Zielinski et al., 2009), a morphological analyzer for German, based on the SMOR-based SFST tools (Schmid et al., 2004).

2.3 GermaNet-based Personalized PageRank

While supervised methods have been extensively shown in the literature to be the best performing ones for monolingual WSD based on a fixed sense inventory, given the unavailability of sense-tagged data for German we need to resort to minimally supervised methods to acquire predominant senses from unlabeled text. Alternatively, we also experiment with extending an existing knowledge-based WSD system to disambiguate German target words in context.

We start by adapting the WSD system from Agirre and Soroa (2009, UKB)², which makes use of a graph-based algorithm, named Personalized PageRank (PPR). This method uses a lexical knowledge base (LKB), e.g. WordNet, in order to rank its vertices to perform disambiguation in context. First, a LKB is viewed as an undirected graph $G = \langle V, E \rangle$ where each vertex $v_i \in V$ represents a concept, e.g. a synset, and each semantic relation between edges, e.g. hypernymy or hyponymy, corresponds to an undirected edge $(v_i, v_j) \in E$. Given an input context $C = \{w_1 \dots w_n\}$, each content word (i.e. noun, verb, adjective or adverb) $w_i \in C$ is inserted in G as a vertex, and linked with directed edges to m associated concepts, i.e. the possible senses of w_i according to the sense inventory of the LKB. Next, the PageRank algorithm (Brin and Page, 1998) is run over the graph G to compute PR , the PageRank score of each concept in the graph given the input context as:

$$PR(v_i) = (1 - d) + d \sum_{j \in \deg(v_i)} \frac{PR(v_j)}{|\deg(v_j)|} \quad (2)$$

where $\deg(v_i)$ is the set of neighbor vertices of vertex v_i , and d is the so-called damping factor (typically set to 0.85). The PageRank score is calculated by iteratively computing Equation 2 for each vertex in the graph, until convergence below a given threshold is achieved, or a fixed number of iterations, i.e. 30 in our case, is executed. While in the standard formulation of PageRank the PR scores are initialized with a uniform distribution, i.e. $\frac{1}{|V|}$, PPR concentrates all initial mass uniformly over the word vertices representing the

¹<http://lucene.apache.org>

²<http://ixa2.si.ehu.es/ukb>

context words in C , in order to compute the structural relevance of the concepts in the LKB given the input context. Finally, given the PageRank scores of the vertices in G and a target word w to be disambiguated, PPR chooses its associated concept (namely, the vertex in G corresponding to a sense of w) with the highest PageRank score.

In order to use UKB to find predominant senses of German words, we first extend it to use GermaNet as LKB resource. This is achieved by converting GermaNet into the LKB data format used by UKB. We then run PPR to disambiguate target words in context within a set of manually annotated test sentences, and select for each target word the sense which is chosen most frequently by PPR for the target word in these sentences.

2.4 System combination

All our methods for predominant sense induction are unsupervised in the sense that they do not require any sense-tagged sentences. However, they all rely on an external resource, namely GermaNet, to provide a minimal amount of supervision. McCarthy et al.’s (2004) method uses the lexical resource to compute the semantic similarity of words. Lapata and Keller (2007) rely instead on its taxonomy structure to expand the sense descriptors of candidate senses based on hypernyms and hyponyms. Finally, UKB uses the full graph of the lexical knowledge base to find structural similarities with the input context.

All these methods include a phase of weak supervision, while in different ways. We thus hypothesize that they are complementary: that is, by combining their sense rankings, we expect their different amounts of supervision to complement each other, thus yielding a better ranking. We accordingly experiment with a simple *majority voting* scheme which, for each target word, collects the predominant senses output by all three systems and chooses the sense candidate with the highest number of votes. In case of ties, we perform a random choice among the available candidates.

3 Experiments and Evaluation

We evaluate the performance of the above methods both on the detection of predominant senses and token-level WSD in context. For this purpose we created a gold standard of sense-annotated sentences following the model of the SensEval evaluation datasets (Section 3.1). The most frequent

sense annotations are then used to provide gold standard predominant senses for German words as the most frequent ones found in the annotated data. Accordingly, we first evaluate our systems in an *intrinsic* evaluation quantifying how well the automatically generated sense rankings model the one from the gold standard sense annotations (Section 3.2). In addition, the gold standard of sense-annotated German words in context is used to *extrinsically* evaluate each method by performing type-based WSD, i.e. disambiguating all contextual occurrences of a target word by assigning them their predominant sense (Section 3.3). Finally, since UKB provides a system to perform token-based WSD, i.e. disambiguating each occurrence of a target word separately, we evaluate its output against the gold standard annotations and compare its performance against the type-based systems.

3.1 Creation of a gold standard

Given the lack of sense-annotated corpora for German in the SensEval and SemEval competitions, we created a gold standard for evaluation, taking the SensEval data for other languages as a model, to ensure comparability to standard evaluation datasets. The construction of our gold standard for predominant sense is built on the hypothesis that the *most frequent sense* encountered in a sample of sentences for a given target word can be taken as the *predominant sense*. While this is arguably an idealization, it follows the assumption that, given balanced data, the predominant sense will be encountered with the highest frequency. In addition, this reflects the standard definition of predominant sense found in WordNet.

We selected the 40 keys from the English SensEval-2 test set³ and translated these into German. In case of alternative translations, the selection took into account part of speech, comparable ambiguity rate, and frequency of occurrence in deWAC (at least 20 sentences). We ensure that the data set reflects the distribution of GermaNet across PoS (the set contains 18 nouns, 16 verbs and 6 adjectives), and yields a range of ambiguity rates between 2 and 25 senses for all PoS. For each target word, we extracted 20 sentences for words with up to 4 senses, and an additional 5 sentences per word for each additional sense. This evalua-

³<http://www.d.umn.edu/~tpederse/data.html>

tion dataset was manually annotated with the contextually appropriate GermaNet senses.

3.2 Modeling human sense rankings from gold standard annotations

Experimental setting. The gold standard ranking of word sense we use is given by frequency of senses in the annotations, i.e. the most frequently annotated word sense represents the predominant one, and so on. For each method, we then evaluate in terms of standard measures of precision (P , the ratio of correct predominant senses to the total of senses output by the system), recall (R , the ratio of correct predominant senses to the total of senses in the gold standard) and F_1 -measure ($\frac{2PR}{P+R}$). Since all methods provide a ranking of word senses, rather than a single answer, we also performed an additional evaluation using the ranking-sensitive metrics of *precision at rank – P@k* i.e. the ratio of correct predominant senses found in the top- k senses to the total of senses output by the system – as well as *Mean Reciprocal Rank – MRR*, namely the average of the reciprocal ranks of the correct predominant senses given by the system.

Results and discussion. Tables 1 and 2 present results for the intrinsic evaluation of the German predominant senses. These are generated based on the methods of McCarthy et al. (2004, MCC), Lapata and Keller (2007, LK), the frequency of sense assignments of UKB to the sense-annotated test sentences, and the system combination (Merged, Section 2.4). In the case of LK, we show for the sake of brevity only results obtained with the best configuration (including counts from PARA, HYPER and HYPOALL with no filtering), as found by manually validating the system output on a held-out dataset of word senses. As a baseline, we use a random sense assignment to find the predominant sense of a word (Random), as well as a more informed method that selects as predominant sense of a word the one whose synset has the largest size (SynsetSize). Each system is evaluated on all PoS and the nouns-only subset of the gold standard.

All systems, except LK on the all-words dataset, perform above both baselines, indicating meaningful output. The drastic performance decrease of LK when moving from nouns only to all PoS is due to the fact that in many cases, i.e. typically for adverbs and adjectives but also for nouns, the GermaNet synsets contain none or few synonyms to construct the base sense descriptions with, as

well as very few hyponyms and hypernyms to expand them (i.e. due to the paucity of connectivity in the taxonomy). Among the available methods, UKB achieves the best performance, since it indirectly makes use of the supervision provided by the words in context. System combination performs lower than the best system: this is because in many cases (i.e. 17 out of 40 words) we reach a tie, and the method randomly selects a sense out of the three available without considering their confidence scores. Following Lapata and Keller (2007), we computed the correlation between the sense frequencies in the gold standard and those estimated by our models by computing the Spearman rank correlation coefficient ρ . In the case of our best results, i.e. UKB, we found that the sense frequencies were significantly correlated with the gold standard ones, i.e. $\rho = 0.44$ and 0.49 , $p << 0.01$, for nouns and all-words respectively.

3.3 Type and token-based WSD for German

Experimental setting. We next follow the evaluation framework established by McCarthy et al. (2004) and Lapata and Keller (2007) and evaluate the sense ranking for the *extrinsic* task of performing WSD on tokens in contexts. We use the sense rankings to tag all occurrences of the target words in the test sentences with their predominant senses. Since such a *type-based* WSD approach only provides a baseline system which performs disambiguation without looking at the actual context of the words, we compare it against the performance of a full-fledged *token-based* system, i.e. UKB.*inst*, which disambiguates each instance of a target word separately based on its actual context.

Results and discussion. The results on the WSD task are presented in Table 3. As in the other evaluation, we use the standard metrics of precision, recall and balanced F-measure, as well as the Random and SynsetSize baselines. We use SynsetSize as a back-off strategy in case no sense assignment is attempted by a system, i.e. similar to the use of the SemCor most frequent sense heuristic for standard English WSD systems. In addition, we compute the performance of the system using the most frequent sense from the test sentences themselves: this represents an oracle system, which uses the most frequent sense from the gold standard to provide an upper-bound for the performance of type-based WSD.

	Nouns only			All words		
	P	R	F ₁	P	R	F ₁
Random	16.67	16.67	16.67	17.50	17.50	17.50
SynsetSize	33.33	33.33	33.33	32.50	32.50	32.50
MCC	44.44	44.44	44.44	35.90	35.00	35.44
LK	56.25	50.00	52.94	29.03	22.50	25.35
UKB	66.67	66.67	66.67	50.00	50.00	50.00
Merged	61.11	61.11	61.11	47.50	47.50	47.50

Table 1: Results against human sense rankings: precision/recall on exact, i.e. predominant-only, senses.

	Nouns only				All words			
	P@1	P@2	P@3	MRR	P@1	P@2	P@3	MRR
baseline	16.67	66.67	77.78	0.50	17.50	52.50	70.00	0.47
SynsetSize	33.33	72.22	88.89	0.61	32.50	55.00	67.50	0.54
MCC	44.44	88.89	88.89	0.69	35.90	66.67	79.49	0.58
LK	56.25	81.25	93.75	0.65	29.03	41.94	48.39	0.29
UKB	66.67	88.89	100.00	0.81	50.00	77.50	87.50	0.68
Merged	61.11	83.33	88.89	0.74	47.50	70.00	70.00	0.59

Table 2: Results against human sense rankings: precision @k and MRR on full sense rankings.

The WSD results corroborate our previous findings from the intrinsic evaluation, namely that: (i) all systems, except LK on the all-words dataset, achieve a performance above both baselines, indicating the feasibility of the task; (ii) the best results on type-based WSD are achieved by selecting the sense which is chosen most frequently by UKB for the target word; (iii) system combination based on a simple majority-voting scheme does not improve the results, due to the ties in the voting and the relative random choice among the three votes. As expected, performing token-based WSD performs better than type-based: this is because, while labeling based on predominant senses represents a powerful option due to the skewness of sense distributions, target word contexts also provide crucial evidence to perform robust WSD.

4 Conclusions and Future Work

We presented a variety of methods to automatically induce resources to perform WSD in German, using GermaNet as a LKB. We applied methods for predominant sense induction, originally developed for English (McCarthy et al., 2004; Lapata and Keller, 2007), to German. We further adapted a graph-based WSD system (Agirre and Soroa, 2009) to label words in context using the GermaNet resource.

	Nouns only P/R/F ₁	All words P/R/F ₁
Random	22.49	15.42
SynsetSize	31.98	24.67
MCC	41.46	27.66
LK	42.28	21.31
UKB	48.78	36.73
Merged	44.72	33.55
UKB. <i>inst</i>	55.49	38.90
Test MFS	64.50	56.54

Table 3: Results for *type*- and *token*-based WSD on the gold standard.

Our results show that we are able to robustly bootstrap baseline systems for the automatic annotation of word senses in German. The systems were evaluated against a carefully created gold standard corpus. Best results were obtained by the knowledge-based system, which profits from its limited supervision by the surrounding context. System integration based on majority voting could not improve over the best system, yielding an overall performance degradation. We leave the exploration of more refined ensemble methods, e.g. weighted voting, to future work.

While our evaluation is restricted to 40 words only, we computed predominant sense rankings for the entire sense inventory of GermaNet. We will make these available in the form of a sense-annotated version of deWAC, as well as an API to access this information in GermaNet.

The present work is *per-se* not extremely novel, but it extends and applies existing methods to create new resources for German. The annotations produced by the WSD systems can serve as a basis for rapid construction of a gold standard corpus by manually validating their output. A natural extension of our approach is to couple it with manual validation frameworks based on crowdsourcing, i.e. Amazon’s Mechanical Turk (cf. Snow et al. (2008)). We leave such exploration to future work.

References

- Eneko Agirre and Philip Edmonds, editors. 2006. *Word Sense Disambiguation: Algorithms and Applications*. Springer.
- Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proc. of EACL-09*, pages 33–41.
- Eneko Agirre, Lluís Márquez, and Richard Wicentowski, editors. 2007. *Proceedings of SemEval-2007*.
- Jordi Atserias, Luis Villarejo, German Rigau, Eneko Agirre, John Carroll, Bernardo Magnini, and Piek Vossen. 2004. The MEANING multilingual central repository. In *Proc. of GWC-04*, pages 80–210.
- Santanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlap as a measure of semantic relatedness. In *Proc. of IJCAI-03*, pages 805–810.
- Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora. *Journal of Language Resources and Evaluation*, 43(3):209–226.
- Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.
- Edward Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proc. of COLING-ACL-06 Interactive Presentation Sessions*, pages 77–80.
- Gerard de Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. In *Proc. of CIKM-09*, pages 513–522.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Iryna Gurevych and Hendrik Niederlich. 2005. Accessing GermaNet data and computing semantic relatedness. In *Comp. Vol. to Proc. of ACL-05*, pages 5–8.
- J. Hall and J. Nivre. 2008. A dependency-driven parser for german dependency and constituency representations. In *Proceedings of the Parsing German Workshop at ACL-HLT 2008*, pages 47–54.
- Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics*.
- Miralla Lapata and Frank Keller. 2007. An information retrieval approach to sense ranking. In *Proc. of NAACL-HLT-07*, pages 348–355.
- Geoffrey Leech. 1992. 100 million words of English: the British National Corpus. *Language Research*, 28(1):1–13.
- Lothar Lemnitzer and Claudia Kunze. 2002. GermaNet – representation, visualization, application. In *Proc. of LREC ’02*, pages 1485–1491.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of COLING-ACL-98*, pages 768–774.
- Mausam, Stephen Soderland, Oren Etzioni, Daniel Weld, Michael Skinner, and Jeff Bilmes. 2009. Compiling a massive, multilingual dictionary via probabilistic inference. In *Proc. of ACL-IJCNLP-09*, pages 262–270.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proc. of ACL-04*, pages 280–287.
- Rada Mihalcea and Phil Edmonds, editors. 2004. *Proceedings of SENSEVAL-3*.
- George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308.
- Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proc. of ACL-10*.
- Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.
- Ted Pedersen and R. Bruce. 1997. Distinguishing word senses in untagged text. In *Proc. EMNLP-97*, pages 197–207.
- Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. MultiWordNet: Developing an aligned multilingual database. In *Proc. of GWC-02*, pages 21–25.

Helmut Schmid, Arne Fitschen, and Ulrich Heid.
 2004. SMOR: A German computational morphology covering derivation, composition and inflection.
 In *Proc. of LREC '04*.

Hinrich Schütze. 1992. Dimensions of meaning. In
Proceedings of Supercomputing '92, Los Alamitos,
 Cal., 16–20 November 1992, pages 787–796.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proc. of EMNLP-08*, pages 254–263.

Dan Tuſiš, Dan Cristea, and Sofia Stamou. 2004. BalkaNet: Aims, methods, results and perspectives. a general overview. *Romanian Journal on Science and Technology of Information*, 7(1-2):9–43.

Piek Vossen, editor. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer, Dordrecht, The Netherlands.

Andrea Zieliński, Christian Simon, and Tilman Wittl.
 2009. Morphisto: Service-oriented open source morphology for German. In Cerstin Mahlow and Michael Piotrowski, editors, *State of the Art in Computational Morphology: Proceedings of the Workshop on Systems and Frameworks for Computational Morphology*, volume 41 of *Communications in Computer and Information Science*, pages 64–75. Springer.

Presupposition Projection and Accommodation in Mathematical Texts

Marcos Cramer Daniel Kühlwein

Mathematisches Institut
University of Bonn
Bonn, Germany

{cramer, kuehlwei}@math.uni-bonn.de bernhard.schroeder@uni-due.de

Bernhard Schröder

Fakultät für Geisteswissenschaften
University of Duisburg-Essen
Essen, Germany

Abstract

This paper discusses presuppositions in mathematical texts and describes how presupposition handling was implemented in the Naproche system for checking natural language mathematical proofs. Mathematical texts have special properties from a pragmatic point of view, since in a mathematical proof every new assertion is expected to logically follow from previously known material, whereas in most other texts one expects new assertions to add logically new information to the context. This pragmatic difference has its influence on how presuppositions can be projected and accommodated in mathematical texts. Nevertheless, the account of presupposition handling developed for the Naproche system turned out to have equivalent projection predictions to an existing account of presupposition projection.

1 Introduction

The special language that is used in mathematical journals and textbooks has some unique linguistic features, on the syntactic, on the semantic and on the pragmatic level: For example, on the syntactic level, it can incorporate complex symbolic material into natural language sentences. On the semantic level, it refers to rigorously defined abstract objects, and is in general less open to ambiguity than most other text types. On the pragmatic level, it reverses the expectation on assertions, which have to be implied by the context rather than adding new information to it.

We call this special language *the semi-formal language of mathematics* (SFLM), and call texts written in this language *mathematical texts*. The content of mathematical texts can be divided into object level content (mathematical statements and

their proofs) and meta level content (e.g. historical remarks and motivation behind certain definitions and one's interest in certain theorems). For the rest of the paper, we will focus on object level content of mathematical texts.

Our work on presuppositions in mathematical texts has been conducted in the context of the *Naproche project*. The Naproche project studies SFLM from the perspectives of linguistics, logic and mathematics. A central goal of Naproche is to develop a controlled natural language (CNL) for mathematical texts and implement a system, the *Naproche system*, which can check texts written in this CNL for logical correctness using methods from computational linguistics and automatic theorem proving.

The Naproche system first translates a CNL text into a *Proof Representation Structure (PRS)*. PRSs are Discourse Representation Structures (DRSs, (Kamp and Reyle, 1993)), which are enriched in such a way as to represent the distinguishing characteristics of SFLM (Cramer et al., 2010). For every statement made in the text, the Naproche System extracts from the PRS a proof obligation that gets sent to an automated theorem prover, in order to check that the claim made by the statement follows from the available information (Cramer et al., 2009).

In this paper we describe how the checking algorithm of Naproche had to be altered when expressions triggering presuppositions were added to the Naproche CNL. Presuppositions also have to be checked for correctness, but behave differently from assertions in mathematical texts because of their special pragmatic properties to be discussed below.

2 Presuppositions

Losely speaking, a *presupposition* of some utterance is an implicit assumption that is taken for granted when making the utterance. In the literature, presuppositions are generally accepted to be triggered by certain lexical items called *presupposition triggers*. Among them are definite noun phrases (in English marked by the definite article “the”, possessive pronouns or genitives), factive verbs (like “regret”, “realize” and “know”), change of state verbs (“stop” and “begin”), iteratives (“again”) and some others.

Presupposition projection is the way in which presuppositions triggered by expressions within the scope of some operator have to be evaluated outside this scope. The precise way in which presuppositions project under various operators has been disputed at great length in the literature (see for example Levinson (1983) and Kadmon (2001) for overviews of this dispute). The DRS-based presupposition projection mechanism that we came up with for dealing with presuppositions in mathematical texts turned out to be equivalent to Heim’s (1983) approach to presupposition projection.

Presupposition accommodation is what we do if we find ourselves faced with a presupposition the truth of which we cannot establish in the given context: We add the presupposition to the context, in order to be able to process the sentence that presupposes it.

In mathematical texts, most of the presupposition triggers discussed in the linguistic literature, e.g. factive verbs, change of state verbs and iteratives, are not very common or even completely absent. Definite descriptions, however, do appear in mathematical texts (e.g. “the smallest natural number n such that $n^2 - 1$ is prime”). And there is another kind of presupposition trigger, which does not exist outside mathematical texts: Function symbols. For example, the devision symbol “/” presupposes that its second (right hand) argument is non-zero; and in a context where one is working only with real and not with complex numbers, the square root symbol “ $\sqrt{}$ ” presupposes that its argument is non-negative. As has been pointed out by Kadmon (2001), the kind of presupposition trigger does, however, not have any significant influence on the projection and accommodation properties of presuppositions. For this reason, we will concentrate on examples with definite de-

scriptions.

Although terminology is not used in a fully uniform fashion among linguists, we will make the following distinctions suitable for our purposes. We analyse noun phrases semantically into a determiner (here: “the”) and a restricting property. Definite noun phrases referring to a single object by a restricting property whose extension contains exactly one object we call *definite descriptions*. Definite noun phrases in the singular with restricting properties whose extension contains more than one object get their referential uniqueness usually by anaphoric reference to an object mentioned previously; they are called *anaphoric definite noun phrases*. A mathematical example of an anaphoric definite noun phrase is “the group” used to refer to a group mentioned recently in the text. The example above (“the smallest natural number n such that $n^2 - 1$ is prime”) was an example of a definite description.

The presupposition of a singular definite description with the restricting property F is that there is a unique object with property F. This presupposition can be divided into two separate presuppositions: One existential presupposition, claiming that there is at least one F, and one uniqueness presupposition, claiming that there is at most one F.

3 Proof Representation Structures

For the purpose of this paper, we provide a simplified definition of Proof Representation Structures, which is very similar to standard definitions of Discourse Representation Structures. A full-fledged definition of Proof Representation Structures can be found in Cramer et al. (2010).

A Proof Representation Structure is a pair consisting of a list of discourse referents and an ordered list of conditions. Just as in the case of DRSs, PRSs and PRS conditions are defined recursively: Let A, B be PRSs and d, d_1, \dots, d_n discourse referents. Then

- for any n -ary predicate p (e.g. expressed by adjectives and noun phrases in predicative use and verbs in SFLM), $p(d_1, \dots, d_n)$ is a condition.
- $\neg A$ is a condition, representing a negation.
- $B \Rightarrow A$ is a condition, representing an assumption (B) and the set of claims made inside the scope of this assumption (A).

d_1, \dots, d_m
c_1
\vdots
c_n

Figure 1: A PRS with discourse referents d_1, \dots, d_m , and conditions c_1, \dots, c_n .

- A is a condition.
- $\text{the}(d, A)$ is a condition, representing a definite description with restricting property F , where d is the discourse referent introduced by this definite description and A is the representation of F .

Apart from the *the*-condition, which was also absent from PRSs as they were defined in Cramer et al. (2010), there are two differences between this definition of PRSs and standard definitions of DRSs: Firstly, the list of PRS conditions is ordered, whereas DRS conditions are normally thought to form an unordered set. Secondly, a bare PRS can be a direct condition of a PRS. Both of these differences are due to the fact that a PRS does not only represent which information is known and which discourse referents are accessible after processing some (possibly partial) discourse, but also represents in which order the pieces of information and discourse referents were added to the discourse context.

Similar to DRSs, we can display PRSs as “boxes” (Figure 1). If $m = 0$, we leave out the top cell.

4 Checking PRSs without presuppositions

In order to explain our treatment of presuppositions, we first need to explain how PRSs without presuppositions are checked for correctness by the Naproche system.

The checking algorithm makes use of *Automated Theorem Provers* (ATPs) for first-order logic (Fitting, 1996).¹ Given a set of *axioms* and a *conjecture*, an ATP tries to find either a proof that the axioms logically imply the conjecture, or build a model for the premisses and the negation of the conjecture, which shows that that they don’t imply

¹The checking algorithm is implemented in such a way that it is easily possible to change the ATP used. Most of our tests of the system are performed with the prover E (Schulz, 2002).

it. With difficult problems, an ATP might not find any proof or counter-model within the time limit that one fixed in advance. A conjecture together with a set of axioms handed to an ATP is called a *proof obligation*.

The checking algorithm keeps a list of first-order formulae considered to be true, called *premises*, which gets continuously updated during the checking process. The conditions of a PRS are checked sequentially. Each condition is checked under the currently active premises. According to the kind of condition, the Naproche system creates obligations which have to be discharged by an ATP.

Below we list how the algorithm proceeds depending on the PRS condition encountered. We use Γ to denote the list of premises considered true before encountering the condition in question, and Γ' to denote the list of premises considered true after encountering the condition in question. A proof obligation checking that ϕ follows from Γ will be denoted by $\Gamma \vdash \phi$. For any given PRS A , we denote by $FI(A)$ the *formula image* of A , which is a list of first-order formulae representing the information introduced in A ; the definitions of $FI(A)$ and of the checking algorithm are mutually recursive, as specified below.

We first present the algorithm for PRS conditions that do not introduce new discourse referents. Next we extend it to conditions introducing discourse referents. In section 5, we extend it further to conditions that trigger presuppositions. (For simplifying the presentation, we treat formula lists as formula sets, i.e. allow ourselves to use set notation when in reality the algorithm works with ordered lists.)

- (1) For a condition of the form $p(d_1, \dots, d_n)$, check $\Gamma \vdash p(d_1, \dots, d_n)$ and set $\Gamma' := \Gamma \cup \{p(d_1, \dots, d_n)\}$.
- (2) For a condition of the form $\neg A$, check $\Gamma \vdash \neg \bigwedge FI(A)$ and set $\Gamma' := \Gamma \cup \{\neg \bigwedge FI(A)\}$.
- (3) For a condition of the form $B \Rightarrow A$, where no discourse referents are introduced in A , check A with initial premise set $\Gamma \cup FI(B)$, and set $\Gamma' := \Gamma \cup \Delta$, where $\Delta := \{\forall \vec{x} (\bigwedge FI(B) \rightarrow \phi) | \phi \in FI(A)\}$ and \vec{x} is the list of free variables in $FI(B)$.
- (4) For a condition of the form A , where no discourse referents are introduced in A , check

A with initial premise set Γ , and set $\Gamma' := \Gamma \cup FI(A)$.

For computing $FI(A)$, the algorithm proceeds analogously to the checking of A , only that no proof obligations are sent to the ATP: The updated premise lists are still computed, and $FI(A)$ is defined to be $\Gamma' - \Gamma$, where Γ is the premise list before processing the first condition in A and Γ' is the premise list after processing the last condition in A . This is implemented by allowing the algorithm to process a PRS A in two different modes: The Check-Mode described above for checking the content of A , and the No-Check-Mode, which refrains from sending proof obligations to the ATP, but still expands the premise list in order to compute $FI(A)$.

For the cases (1) and (2), it is easy to see that what gets added to the list of premises in the Check-Mode is precisely what has been checked to be correct by the ATP. In the cases (3) and (4), it can also be shown that what gets added to the set of premises has implicitly been established to be correct by the ATP.

Special care is required when in conditions of the form $B \Rightarrow A$ or A , new discourse referents are introduced in A . Let us first consider the simpler case of a condition of the form A that introduces new discourse referents; this corresponds to sentences like “There is an integer x such that $2x - 1$ is prime.”, i.e. sentences with an existential claim, which make the existentially introduced entity anaphorically referencible by the later discourse. As would be expected, we need to check $\Gamma \vdash \exists x(\text{integer}(x) \wedge \text{prime}(2x - 1))$ in this case. But we cannot just add $\exists x(\text{integer}(x) \wedge \text{prime}(2x - 1))$ to the premise set, since the first order quantifier \exists does not have the dynamic properties of the natural language quantification with “there is”: When we later say “ $x \neq 1$ ”, the proof obligation of the form $\Gamma \cup \{\exists x(\text{integer}(x) \wedge \text{prime}(2x - 1))\} \vdash x \neq 1$ for this sentence would not make sense, since the free x in the conjecture would not corefer with the existentially bound x in the axiom.

We solve this problem by *Skolemizing* (Brachman and Levesque, 2004) existential formulae before adding them to the premise list: In our example, we add $\text{integer}(c) \wedge \text{prime}(2c - 1)$ (for some new constant symbol c) to the premise list. Later uses of x will then also have to be substituted by c , so the proof obligation for “ $x \neq 1$ ” becomes $\Gamma \cup \{\text{integer}(c) \wedge \text{prime}(2c - 1)\} \vdash c \neq 1$.

Given that the discourse referents introduced in A become free variables in $FI(A)$, we can require more generally: For a condition of the form A which introduces discourse referents, check $\Gamma \vdash \exists \vec{x}(\bigwedge FI(A))$ (where \vec{x} is the list of free variables in $FI(A)$), and set $\Gamma' := \Gamma \cup S(FI(A))$. We define $S(FI(A))$ (the Skolemized version of $FI(A)$) to be the set of formulae that we get when we substitute each free variable used in some formula in $FI(A)$ by a different new constant symbol, ensuring that the same constant symbol is used for the same free variable across different formulae in $FI(A)$.

In the case of conditions of the form $B \Rightarrow A$ with A introducing new discourse referents, we need the more general kind of Skolemization, which involves introducing new function symbols rather than new constant symbols: We proceed in the same way as for the case when A doesn't introduce new discourse referents, only that in the definition of Γ' we replace Δ by its Skolemized form $S(\Delta)$. $S(\Delta)$ consists of Skolemized versions of the formulae in Δ , where the Skolem functions are chosen in such a way that any free variable appearing in more than one formula in Δ gets replaced by the same function across the different formulae in which it appears.

5 Checking PRSs with presuppositions

Most accounts of presupposition make reference to the *context* in which an utterance is uttered, and claim that presuppositions have to be satisfied in the context in which they are made. There are different formalisations of how a context should be conceptualised. For enabling the Naproche checking algorithm described in the previous section to handle presuppositions, it is an obvious approach to use the list of active premises (which include definitions) as the context in which our presuppositions have to be satisfied.

As noted before, assertions in mathematical texts are expected to be logically implied by the available knowledge rather than adding something logically new to it. Because of this pragmatic peculiarity, both presuppositions and assertions in proof texts have to follow logically from the context. For a sentence like “The largest element of M is finite” to be legitimately used in a mathematical text, both the unique existence of a largest element of M and its finiteness must be inferable

from the context.²

This parallel treatment of presuppositions and assertions, however, does not necessarily hold for presupposition triggers that are subordinated by a logical operation like negation or implication. For example, in the sentence “*A* does not contain the empty set”, the existence and uniqueness presuppositions do not get negated, whereas the containment assertion does. This is explained in the following way: In order to make sense of the negated sentence, we first need to *make sense of* what is inside the scope of the negation. In order to make sense of some expression, all presuppositions of that expression have to follow from the current context. The presuppositions triggered by “the empty set” are inside the scope of the negation, so they have to follow from the current context. The containment assertion, however, does not have to follow from the current context, since it is not a presupposition, and since it is negated rather than being directly asserted.

In our implementation, *making sense of something* corresponds to processing its PRS, whether in the Check-Mode or in the No-Check-Mode. So according to the above explanation, presuppositions, unlike assertions, also have to be checked when encountered in the No-Check-Mode.

For example, the PRS of sentence (5) is (6).

(5) *A* does not contain the empty set.

(6)	$\neg \text{the}(x, \boxed{\text{empty}(x)} \wedge \boxed{\text{set}(x)}) \rightarrow \neg \text{contain}(A, x)$
-----	--

When the checking algorithm encounters the negated PRS, it needs to find the formula image of the PRS, for which it will process this PRS in No-Check-Mode. Now the *the*-condition triggers two presuppositions, which have to be checked despite being in No-Check-Mode. So we send the proof obligations (7) and (8) (for a new constant symbol *c*) to the ATP. Finally, the proof obligation that we want for the assertion of the sentence is (9).

(7) $\Gamma \vdash \exists x (\text{empty}(x) \wedge \text{set}(x))$

²The remaining distinctive feature between assertions and presuppositions is that the failure of the latter ones makes the containing sentences meaningless, not only false.

(8) $\Gamma \cup \{\text{empty}(c) \wedge \text{set}(c)\} \vdash \forall y (\text{empty}(y) \wedge \text{set}(y) \rightarrow y = c)$

(9) $\Gamma \cup \{\text{empty}(c) \wedge \text{set}(c), \forall y (\text{empty}(y) \wedge \text{set}(y) \rightarrow y = c)\} \vdash \neg \text{contain}(A, c)$

In order to get this, we need to use the non-presuppositional formula image $\{\text{contain}(A, c)\}$ of the negated PRS: The non-presuppositional formula image is defined to be the subset of formulae of the formula image that do not originate from presuppositions. When implementing the checking algorithm for PRSs with presuppositions, we have to use this non-presuppositional formula image wherever we used the formula image in the original checking algorithm. The presupposition premises which get pulled out of the formula image have to be added to the list of premises that were active before starting to calculate the formula image.

This pulling out of presupposition premises is not always as simple as in the above example. Consider for example sentence (10), whose PRS is (11).

(10) There is a finite non-empty set *M* of natural numbers such that the largest element of *M* is even.³

(11)	\boxed{M} $\text{finite}(M)$ $\text{non-empty}(M)$ $\text{set_of_nats}(M)$ $\text{the}(x, \boxed{\text{largest_elt}(x, M)})$ $\text{even}(x)$
------	---

The Skolemized premise from the existential presupposition is *largest_elt(c, M)*, which contains a free occurrence of the variable *M*, but should be pulled out of the PRS introducing *M*, i.e. out of the scope of *M*, in order to be added to the set Γ of premises available before encountering this

³The definite noun phrase “The largest element of *M*” can be read like a function depending on *M*. When, like in our example, such functional definite descriptions are used as functions on a variable that we are quantifying over, the presuppositions of the functional definite description can restrict the domain of the quantifier to entities for which the presupposition is satisfied. Such a restriction of a quantifier is an instance of accommodation (local accommodation in our account), which will be treated in section 7. In this section we are interested in presupposition handling without accommodation, i.e. without restricting the domain of the quantifier in this example. So the presuppositions of “the largest element of *M*” have to be fulfilled for any finite non-empty set *M* of natural numbers.

sentence. Pulling this occurrence of M out of the scope of M would make the premise meaningless, so we need a more sophisticated approach to pulling out presupposition premises:

According to the above account, we will check the existential presupposition in question using the proof obligation (12). Given that M does not appear in Γ (as it is a newly introduced discourse referent), this is logically equivalent to having checked (13), whose Skolemized form (14) will be added to Γ (where sk_x is the new function symbol introduced for x when Skolemizing). This extended premise set is used to check the existential claim of the sentence in (15).

- (12) $\Gamma \cup \{finite(M), non-empty(M), set_of_nats(M)\} \vdash \exists x largest_elt(x, M)$
- (13) $\Gamma \vdash \forall M (finite(M) \wedge non-empty(M) \wedge set_of_nats(M) \rightarrow \exists x largest_elt(x, M))$
- (14) $\forall M (finite(M) \wedge non-empty(M) \wedge set_of_nats(M) \rightarrow largest_elt(sk_x(M), M))$
- (15) $\Gamma \cup \{(14)\} \vdash \exists M (finite(M) \wedge non-empty(M) \wedge set_of_nats(M) \wedge even(sk_x(M)))$

6 Comparison to Heim's presupposition projection

Heim (1983) is concerned with the projection problem, i.e. with “predicting the presuppositions of complex sentences in a compositional fashion from the presuppositions of their parts”. For us, the projection problem only had indirect importance: The reason for our occupation with presupposition was to be able to check mathematical texts containing presupposition triggers for correctness. This does involve checking that the presuppositions of every trigger are satisfied in the local context of the trigger, but it doesn't necessarily involve explicitly computing presuppositions for complex sentences.

Given the sentence (16), Heim's theory predicts that the existential presupposition of the definite description gives rise to the presupposition (17) for the complex sentence.

- (16) If x is positive, then the multiplicative inverse of x is positive.
- (17) If x is positive, then x has a multiplicative inverse.

$$(18) \quad \boxed{\begin{array}{c} x \\ pos(x) \end{array}} \Rightarrow \boxed{\begin{array}{c} the(y, \boxed{mult_inv(y, x)}) \\ pos(y) \end{array}}$$

In our treatment of presupposition, computing the presupposition (17) explicitly is not the central issue; what we do instead is to justify the presupposition with the information locally available when encountering the presupposition. In the example sentence, whose PRS is (18), the formula image of the left PRS of the implication condition ($\{pos(x)\}$) is Skolemized and added to the set Γ of premises available before encountering this sentence, so that the set of premises available when encountering the *the*-condition is $\Gamma \cup \{pos(c)\}$. Hence the proof obligation for justifying the existential presupposition of the definite description is (19). Having checked this is equivalent to having checked (20), i.e. having deduced Heim's projected presupposition from Γ .

- (19) $\Gamma \cup \{pos(c)\} \vdash \exists y mult_inv(y, c)$
- (20) $\Gamma \vdash \forall x (pos(x) \rightarrow \exists y mult_inv(y, x))$

Also for presuppositions subordinated under other kinds of logical operators, our theory is in this way equivalent to Heim's theory. This is the sense in which one can say that our theory is equivalent to Heim's theory.⁴ On the other hand, we arrive at these results in a somewhat different way to Heim: Heim defines contexts in a semantical way as sets of possible worlds or, in her account of functional definite descriptions, as a set of pairs of the form $\langle g, w \rangle$, where g is a sequence of individuals and w is a possible world, whereas we define a context syntactically as a list of first-order formulae.

7 Accommodation

One commonly distinguishes between *global* and *local* accommodation. Global accommodation is the process of altering the global context in such

⁴Here we are comparing our theory without accommodation to Heim's theory without accommodation. Heim calls the projected universal presupposition for functional noun phrases (as discussed in the previous section, cf. example (10) with presupposition (13)) “unintuitively strong”, and gives an explanation for this using accommodation. But this universal presupposition is what her account without accommodation predicts, just as in our case. Cf. the justification for the strong universal presupposition in footnote 3.

a way that the presupposition in question can be justified; local accommodation on the other hand involves only altering some local context, leaving the global context untouched. It is a generally accepted pragmatic principle that *ceteris paribus* global accommodation is preferred over local accommodation.

In the introduction, we mentioned the pragmatic principle in mathematical texts that new assertions do not add new information (in the sense of logically not infereable information) to the context. Here “context” of course doesn’t refer to our formal definition of context as a list of formulae. In fact, if we take a possible world to be a situation in which certain axioms, definitions, and assumptions hold or do not hold, we can make sense of the use of “context” in this assertion by applying Heim’s definition of a context as a set of possible worlds. When mathematicians state axioms, they limit the context, i.e. the set of possible worlds they consider, to the set where the axioms hold. Similarly, when they make local assumptions, they temporarily limit the context. But when making assertions, these assertions are thought to be logically implied by what has been assumed and proved so far, so they do not further limit the context.

This pragmatic principle of not adding anything to the context implies that global accommodation is not possible in mathematical texts, since global accommodation implies adding something new to the global context. Local accommodation, on the other hand, is allowed, and does occur in real mathematical texts:

Suppose that f has n derivatives at x_0 and n is the smallest positive integer such that $f^{(n)}(x_0) \neq 0$.

(Trench, 2003)

This is a local assumption. The projected existential presupposition of the definite description “the smallest positive integer such that $f^{(n)}(x_0) \neq 0$ ” is that for any function f with some derivatives at some point x_0 , there is a smallest positive integer n such that $f^{(n)}(x_0) \neq 0$. Now this is not valid in real analysis, and we cannot just assume that it holds using global accommodation. Instead, we make use of local accommodation, thus adding the accommodated fact that there is a smallest such integer for f to the assumptions that we make about f with this sentence.

The fact that one has to accommodate locally rather than globally does not, however, always fix which context we alter when accommodating. Consider for example sentence (21), used in a context where we have already defined a set A_x of real numbers for every real number x .

- (21) For all $x \in \mathbb{R}$, if A_x doesn’t contain $\frac{1}{x}$, then A_x is finite.

The question is whether we need to check the finiteness of A_0 in order to establish the truth of (21), or whether the finiteness of A_0 is irrelevant. Since the use of $\frac{1}{x}$ presupposes that $x \neq 0$, which doesn’t hold for any arbitrary $x \in \mathbb{R}$, we have to locally accommodate that $x \neq 0$. But we can either accommodate this within the scope of the negation or outside the scope of the negation, but still locally within the conditional. In the first case, we have to establish that A_0 is finite, whereas in the second case we don’t.

Unlike the presupposition handling described in the previous sections, local accommodation has not yet been implemented into Naproche. Before this can be done, we need some mechanism for deciding which of a number of possible local accommodations is preferred in cases like the above.

8 Related Work

Presuppositions in mathematical texts have already been studied before: Zinn (2000) discusses presuppositions and implicatures in mathematical texts. His work on presuppositions focuses on the presuppositions that are justified using information from proof plans. Since Naproche currently doesn’t use proof plans, this kind of presupposition is not yet implemented in the Naproche CNL.

Ganesalingam (2009) describes an innovative way of computing the presuppositions triggered by mathematical function symbols (like $-^{-1}$) and the presuppositions given rise to by selective restrictions (e.g. the presupposition “ x is a natural number” of the utterance “ x is prime”) from the definitions where the corresponding function symbols or expressions were defined. Once Naproche implements other presupposition triggers than just definite descriptions, an algorithm similar to that presented by Ganesalingam will be implemented for computing presuppositions triggered by symbols or expressions defined in the text.

9 Conclusion

In this paper we discussed the handling of presuppositions in the checking algorithm of the Naproche system, and compared its projection predictions to those of Heim (1983). We noted that our projection predictions are equivalent to those of Heim, despite the fact that we arrive at these predictions in a different way.

Additionally, we considered accommodation in mathematical texts, and noted that global accommodation is blocked by a pragmatic peculiarity of mathematical texts. Future work will involve implementing local accommodation into the Naproche system.

Zinn, Claus. 2000. *Computing Presuppositions and Implicatures in Mathematical Discourse*. J. Bos and M. Kohlhase (eds.). *Proceedings of the Second Workshop on Inference in Computational Semantics*, 121-135.

References

- Brachman, Ronald J., and Hector J. Levesque. 2004. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, Massachusetts, US.
- Cramer, Marcos, Peter Koepke, Daniel Kühlwein, and Bernhard Schröder. 2009. *The Naproche System. Calculemus 2009 Emerging Trend Paper*.
- Cramer, Marcos, Bernhard Fisseni, Peter Koepke, Daniel Kühlwein, Bernhard Schröder, and Jip Veldman. 2010 (in press). *The Naproche Project – Controlled Natural Language Proof Checking of Mathematical Texts. CNL 2009 Workshop, LNAI 5972 proceedings*. Springer.
- Fitting, Melvin. 1996. *First-order logic and automated theorem proving*. Springer. Springer.
- Ganesalingam, Mohan. 2009. *The Language of Mathematics*. Doctoral thesis draft. <http://people.pwf.cam.ac.uk/mg262/GanesalingamMdis.pdf>.
- Heim, Irene. 1983. *On the projection problem for presuppositions*. D. Flickinger et al. (eds.). *Proceedings of the Second West Coast Conference on Formal Linguistics*, 114-125.
- Kamp, Hans, and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language*. Kluwer Academic Publisher.
- Kadmon, Nirit. 2001. *Formal Pragmatics*. Wiley-Blackwell, Oxford, UK.
- Levinson, Stephen C. 1983. *Pragmatics*. Cambridge University Press, Cambridge, UK.
- Schulz, Stephan. 2002. E – A Brainiac Theorem Prover. *Journal of AI Communications*, 15(2):111–126.
- Trench, William F. 2003. *Introduction to Real Analysis*. Pearson Education.

Inkrementelle Koreferenzanalyse für das Deutsche

Manfred Klenner
 Institut für Computerlinguistik
 Universität Zürich
 Schweiz
 {klenner, tuggener}@cl.uzh.ch

Don Tuggener
 Institut für Computerlinguistik
 Universität Zürich
 Schweiz
 {klenner, tuggener}@cl.uzh.ch

Angela Fahrni
 HITS gGmbH
 Heidelberg
 Deutschland
 Angela.Fahrni@h-its.org

Abstract

Es wird ein inkrementeller Ansatz zur Koreferenzanalyse deutscher Texte vorgestellt. Wir zeigen anhand einer breiten empirischen Untersuchung, dass ein inkrementelles Verfahren einem nicht-inkrementellen überlegen ist und dass jeweils die Verwendung von mehreren Klassifizierern bessere Resultate ergibt als die Verwendung von nur einem. Zudem definieren wir ein einfaches Salienzmass¹, dass annähernd so gute Ergebnisse ergibt wie ein ausgefeiltes, auf maschinellem Lernen basiertes Verfahren. Die Vorverarbeitung erfolgt ausschliesslich durch reale Komponenten, es wird nicht – wie so oft – auf perfekte Daten (z.B. Baumbank statt Parser) zurückgegriffen. Entsprechend tief sind die empirischen Ergebnisse. Der Ansatz operiert mit harten linguistischen Filtern, wodurch die Menge der Antezedenskandidaten klein gehalten wird. Die Evaluierung erfolgt anhand der Koreferenzannotationen der TüBa-D/Z.

1 Einleitung

Empirische Untersuchungen zur Koreferenzresolution gehen oft von Vereinfachungen aus. Die gängigsten Idealisierungen sind:

- Ausklammerung der Anaphorizitätsentscheidung (*true mentions only*)
- Nutzung einer Baumbank
 - perfekte Bäume
 - perfekte funktionale Information (grammatische Funktionen)
 - perfekte morphologische Kategorisierungen

¹Die Schweizer Rechtschreibung kennt kein scharfes s.

Oft wird von sog. *true mentions* ausgegangen, man verwendet also nur diejenigen Nominalphrasen (NPen), die gemäss Gold Standard in einer Koreferenzmenge sind. Die meisten nicht-pronominalen Nominalphrasen sind aber nicht-anaphorisch. Das Problem, diese von den anaphorischen zu unterscheiden, wird ausgeklammert. Empirische Werte unter dieser Setzung liegen etwa 15% zu hoch, vgl. (Klenner & Ailloud, 2009).

Es wird heutzutage nahezu ausschliesslich mit statistischen Methoden (inkl. *machine learning*) gearbeitet. Anaphernresolution wird dann meist als paarweise Klassifikationsaufgabe aufgefasst. Es werden unter anderem folgende Merkmale (*features*) beim Trainieren eines supervisierten Klassifizierers verwendet: die grammatischen Funktionen der NPen, ihre Einbettungstiefe im Syntaxbaum, die Wortklasse und die morphologischen Merkmale der Köpfe der NPen. Falls, wie im Falle der TüBa-D/Z, eine Baumbank mit diesen Spezifikationen vorhanden ist, ist die Versuchung gross, auf reale Komponenten zu verzichten und diese Goldstandardinformation zu nutzen.

Man findet auch wirklich kaum ein System, das vollständig auf realen Modalitäten beruht, d.h. das diese Merkmale mit einem Parser und einer Morphologie zu bestimmen versucht. Die Frage, wo wir heute bei der Lösung der Probleme im Bereich der Koreferenzresolution stehen, ist daher nicht einfach zu beantworten. Es ist deswegen auch schwierig, verschiedene Systeme zu vergleichen: Wir (die Autoren dieses Beitrags eingeschlossen) idealisieren gewissermassen aneinander vorbei. Dabei sind mittlerweile auch für das Deutsche Chunker und Parser verfügbar (z.B. Sennrich et al., 2009).

Der vorliegende Beitrag ist vor allem dazu gedacht, eine Baseline für das Deutsche aufzustellen, die ausschliesslich auf realen Vorverarbeitungskomponenten beruht: Gertwol, GermaNet, Wortschatz Leipzig, Wikipedia und einem Dependenzparser, der auf einer manuell konstruierten Grammatik basiert, jedoch eine statistische Disambiguierung durchführt. Wir haben verschiedene empirische Szenarien durchgespielt und sind zu dem Ergebnis gekommen, dass ein inkrementelles, mit harten Filtern operierendes System zur Koreferenzanalyse (Nominal- und Pronominalanaphern) die besten Ergebnisse liefert. Zur paarweisen Klassifikation der Antezedens-Anapher-Kandidaten verwenden wir TiMBL, ein ähnlichkeitsbasiertes Lernverfahren. Wir zeigen, dass ein einzelner Klassifizierer schlechtere Ergebnisse liefert als der Einsatz von vielen, auf die einzelnen Anapherarten (Personal-, Possessiv-, Relativ- und Reflexivpronomen, sowie Nomen) zugeschnittenen Klassifizierern. Dies gilt für beide Varianten: die nicht-inkrementelle Variante und die inkrementelle (die mit einer Client-Server-Architektur realisiert ist). Wir experimentieren mit verschiedenen Merkmalen, wie sie in der Literatur diskutiert werden. Darüberhinaus erlaubt unser inkrementelles Modell aber neue, koreferenzmengenbezogene Merkmale.

Eigentlich dazu gedacht unsere Baseline zu definieren, stellte sich das von uns empirisch definierte Salienzmass als beinahe ebenbürtige, vor allem aber wesentlich einfachere und schnellere Alternative zu dem Einsatz eines Klassifizierer heraus.

Eine Demoversion unseres inkrementellen Systems ist unter <http://kitt.cl.uzh.ch/kitt/cores/> verfügbar.

2 Modelle zur Koreferenzresolution

Was ist das beste Modell zur Koreferenzanalyse? Inkrementell oder nicht-inkrementell; ein Klassifizierer oder viele Klassifizierer; salienzbasiert oder mittels maschinellem Lernen?

3 Salienz

Zur Modellierung von Salienz existiert eine Reihe unterschiedlicher, aber miteinander verwandter Ansätze. (Lappin & Leass, 1994) legen manuell Gewichte für grammatischen Funktionen fest (das Subjekt erhält den höchsten Wert) und begünstigen neben kurzer Entfernung zwischen

Antezedenskandidat und Anapher die Parallelität von grammatischen Funktionen. Unser Mass (siehe weiter unten) kann als eine empirische Variante dieser Idee interpretiert werden. (Mitkov, 1998) adaptiert und erweitert das Verfahren. Er modelliert z.B. mit dem Givenness-Merkmal das Thema-Rhema-Schema (z.B. ob eine NP die erste im Satz und somit Thema ist). Weiter wird u.a. bestimmt, ob eine NP im Titel oder in einer Überschrift vorkommt (Section Heading Preference), wie oft sie im Text vorhanden ist (Lexical Reiteration) und ob sie von einer Menge bestimmter Verben fokussiert wird (Indicating Verbs). Auch die Definitheit der NP wird berücksichtigt.

Seit diesen Pionierarbeiten hat sich in Bezug auf die Salienzmodellierung nicht mehr viel getan. Vielmehr wurde mit der Auswertung und Aufsummierung der Salienzwerte experimentiert (z.B. statistische Verfahren etwa Ge et al., 1998). Später wurden die Merkmale in Ansätze mit maschinellem Lernen übernommen.

Wir haben mit einem rein korpusbasiertem Salienzmass experimentiert, wobei die Salienz einer NP durch die Salienz der grammatischen Funktion gegeben ist, die die NP innehaltet. Die Salienz einer grammatischen Funktion GF ist definiert als die Anzahl der Anaphern, die als GF geparsert wurden, dividiert durch die Gesamtzahl der Anaphern. Es handelt sich also um folgende bedingte Wahrscheinlichkeit:

$$P(Y \text{ realisiert GF} | Y \text{ ist eine Anapher}).$$

Insgesamt 13 grammatische Funktionen (oder Dependenzlabel) erhalten so eine Salienz, darunter: Adverbiale (adv), Appositionen (appo), direkte und indirekte Objekte, das Subjekt. Wie erwartet ist das Subjekt mit 0.106 am salientesten, gefolgt vom direkten Objekt.

Unser Salienzmass kann in Verbindung mit unserem inkrementellen Modell folgenderweise verwendet werden: Für jedes Pronomen wird das salienteste Antezedens ermittelt. Dieses ist über die Salienz der grammatischen Funktion des Antezedenskandidaten und – bei gleicher Funktion – über seine Nähe zum Pronomen (i.e. der Anapher) eindeutig und sehr schnell bestimmbar. Für Nominalanaphern eignet sich dieses Mass nur bedingt, da Nomen im Gegensatz zu den meisten Pronomen ja nicht in jedem Fall anaphorisch sind (Salienz kennt aber keinen Schwellenwert, was zur Auflösung viel zu vieler Nomen führen würde).

3.1 Nicht-inkrementelles Modell

Die meisten Ansätze zur Koreferenzresolution sind sequentiell: erst wird ein Klassifizierer trainiert, dann werden alle (Test-)Paare gebildet und dem Klassifizierer in einem Schritt zur Klassenbestimmung (anaphorisch, nicht-anaphorisch) vorgelegt. In einem solchen nicht-inkrementellen, paarweisen Verfahren sind alle Entscheidungen lokal – jede Entscheidung ist völlig unabhängig von bereits getroffenen Entscheidungen. Dies führt u.a. dazu, dass nachträglich ein Koreferenzclustering durchgeführt werden muss, bei dem alle Paare in Koreferenzmengen zusammengeführt werden und nach Möglichkeit Inkonsistenzen beseitigt werden. Denn es entstehen (nahezu unvermeidbar) z.B. nicht kompatible Zuordnungen von transitiv verknüpften Ausdrücken. So ist in der Sequenz 'Hillary Clinton .. sie .. Angela Merkel' jeder Name mit dem 'sie' kompatibel, die beiden Namen selbst hingegen nicht (da zwei nicht-matchende Eigennamen i.d.R. nicht koreferent sein können). Trifft ein lokal operierender Klassifizierer für jedes Pronomen-Namen-Paar jedoch eine positive Entscheidung, entsteht via Transitivity eine Inkonsistenz (H. Clinton und A. Merkel sind dann transitiv koreferent). Diese muss das nachgeschaltete Clustering beseitigen, indem eine der beiden Koreferenzentscheidungen (der beiden Pronomen-Namen-Paare) rückgängig gemacht wird. Es gibt verschiedene Clusteringansätze: *best-first* (das wahrscheinlichste Antezedens), *closest-first* (das am nächsten liegende), *aggressive merging* (alle positiven werden verknüpft). Man kann diesen Clusteringschritt aber auch als Optimierungsproblem auffassen, vgl. (Klenner, 2007) und (Klenner & Ailloud, 2009), wobei der Klassifizierer Gewichte liefert und linguistische motivierte Constraints die Optimierungsschritte beschränken. Unser Clusteringverfahren arbeitet mit einem Algorithmus aus dem Zero-One Integer Linear Programming, dem Balas-Algorithmus (Balas, 1965). Dabei werden die Antezedens-Anapher-Kandidaten aufsteigend nach ihrem Gewicht geordnet (Minimierung der Kosten ist das Ziel) und von links nach rechts aufgelöst: solange keine Constraints verletzt werden, wird jedes Paar mit Kosten kleiner 0.5 koreferent gesetzt. Verglichen mit einem *aggressive merging* gewinnt man so 2-4 % F-Mass.

Ein weiteres Problem der paarweisen Klassifizierung (ob inkrementell oder nicht-inkrementell) ist das Fehlen globaler Kontrolle. Obwohl z.B. Possessivpronomen in jeden Fall anaphorisch sind (Ausnahmen sind in der TüBa-D/Z sehr rar), kann man dies dem Klassifizierer nicht als Restriktion jeder gültigen Lösung vorschreiben. Es tritt sehr häufig der Fall ein, dass Promonen vom Klassifizierer kein Antezedens zugewiesen bekommen (d.h. kein Paar kommt über die Schranke von 0.5%). Dies kann man im nicht-inkrementellen Modell nachträglich durch eine Forcierung von Koreferenz reparieren, indem man den besten (d.h. am wenigsten) negativen Kandidaten als Antezedens nimmt. Im Falle eines inkrementellen Modells kann eine solche Bindungsforderung direkt eingelöst werden.

3.2 Inkrementelles Modell

Im Gegensatz zum nicht-inkrementellen Ansatz sind bei einem inkrementellen Ansatz die entstehenden Koreferenzmengen sofort verfügbar, Klassifikationsentscheidungen werden nicht auf einzelne Antezedenskandidaten, sondern auf die gesamte Koreferenzmenge, bzw. einen prototypischen Stellvertreter bezogen. So etwa im obigen Beispiel: entscheidet der Klassifizierer beispielsweise, dass 'sie' eine Anapher zu 'Hillary Clinton' ist, also die Koreferenzmenge [Hillary Clinton, sie] eröffnet wird, dann wird die NP 'Angela Merkel' nicht wie im nicht-inkrementellen Fall mit 'sie' verglichen, sondern mit 'Hillary Clinton' oder gar einem virtuellen Stellvertreterobjekt der Koreferenzmenge, das die Eigenschaften der ganzen Koreferenzmenge repräsentiert. So liefert 'Hillary Clinton' eine semantische Restriktion (Person, weiblich), aber keine morphologische. Obgleich 'sie' morphologisch ambig ist (z.B. Singular und Plural), kann es im Zusammenspiel mit der Information 'weiblich' auf den Singularfall in der dritten Person restriktiert werden.

Ein weiteres Problem nicht-inkrementeller Ansätze ist, dass zu viele negative Beispiele generiert werden (vgl. (Wunsch et al., 2009) wo dieses Problem mittels Sampling gelöst werden soll). Dies führt zu einer Verzerrung des Klassifizierers, er erwirbt eine Präferenz zur negativen Klassifikation. Auch dies kann mit einem inkrementellen Modell abgemildert werden, denn pro Koreferenzmenge muss nur einmal verglichen werden; die restlichen Elemente der

```

1   for i=1      to laenge(I)
2       for j=1 to laenge(C)
3           rj := repräsentatives, legitimes Element der Koreferenzmenge Cj
4           Cand := Cand ⊕ rj if kompatibel(rj, mi)
5       for k= laenge(P) to 1
6           pk:= k-tes, legitimes Element des Puffers
7           Cand := Cand ⊕ pk if kompatibel(pk, mi)
8   if Cand = {} then P := P ⊕ mi
9   if Cand ≠ {} then
10      ordne Cand nach Salienz oder Gewicht
11      b := dasjenige e aus Cand mit dem höchsten Gewicht
12      C := erweitere(C,b,mi)

```

Figure 1: Inkrementelle Koreferenzresolution: Basisalgorithmus

Koreferenzmenge sind nicht erreichbar. Dies reduziert die Menge der Paare insgesamt – sowohl der positiven als auch der negativen (siehe die Verhältnisangaben im Abschnitt 'Experimente').

Die Paargenerierung wird durch Filter restriktiert. Neben den naheliegenden morphologischen Bedingungen (z.B. Person-, Numerus- und Genuskongruenz bei Personalpronomen), gibt es semantische Filter basierend auf GermaNet und Wortschatz Leipzig bei den Nominalanaphern. Die semantischen Filter sind sehr restriktiv, so dass viele 'false negatives' entstehen, was eine recht tiefe Obergrenze (*upper bound*) generiert (ein F-Mass von 75.31%, eine Präzision von 81.58% und eine Ausbeute von 69.95%). Zwei Nomen sind semantisch kompatibel, wenn sie synonym sind, oder eines das Hyperonym des anderen ist. Nicht-kompatible Paare werden ausgesondert (das Prinzip von harten Filtern). Unsere Experimente haben gezeigt, dass restriktive Filter besser sind als laxer oder gar keine Filter.

Abbildung (Figure) 1 gibt den Basisalgorithmus zur inkrementellen Koreferenzresolution wieder. Seien I die chronologisch geordnete Liste von Nominalphrasen, C die Menge der Koreferenzmengen und P ein Puffer, in dem NPs gesammelt werden, die nicht anaphorisch sind (aber vielleicht als Antezedens in Frage kommen); m_i sei die aktuelle NP und ⊕ repräsentiere Listenverketzung (genauer 'Hinzufügen eines Elements'). Für jede NP werden anhand der existierenden Koreferenzpartition und dem Puffer (einer Art Warteliste) Kandidaten (Cand) generiert (Schritte 2-7), denen dann entweder von einem Klassifizierer oder über die Salienz ihrer grammatischen Funktion ein Gewicht zugewiesen wird (Schritt 10). Der

Antezedenskandidat b mit dem höchsten Gewicht wird ausgewählt und die Koreferenzpartition wird um m_i erweitert (Schritt 11 und 12). Je nachdem, was b ist, heisst das, dass die Koreferenzmenge von b um m_i erweitert wird, oder dass eine neue Koreferenzmenge bestehend aus m_i und b eröffnet wird. Falls keine Kandidaten gefunden wurden, wird m_i gepuffert, da es zwar selbst nicht anaphorisch ist, aber als Antezedens für nachfolgende NPs verfügbar sein muss (Schritt 8). Pronomen und (normale) Nomen müssen in einem Fenster von 3 Sätzen gebunden werden (dies ist die Bedeutung von 'legitim' in den Zeilen 3 und 6), Eigennamen können auch weiter zurück (auf Eigennamen) referieren (auch durch diesen Filter werden 'false negatives' produziert und auch hier gilt, dass ein liberales Setting zu schlechteren empirischen Werten führt).

Die Kompatibilität zweier NPs (Schritte 4 bzw. 7) ist POS-spezifisch. Zwei Personalpronomen müssen z.B. im Numerus, Genus und Person kongruieren, während zwei Nomen nur im Numerus übereinstimmen müssen ('der Weg' .. 'die Strecke'), jedoch semantisch kompatibel sein müssen. Im Moment beschränkt sich dies auf eine GermaNet-Abfrage (Synonyme und Hyponyme sind erlaubt) und Wortschatz Leipzig (Synonyme).

Mit Blick auf die Paargenerierung (beim Machine Learning) lässt sich sagen: Die Anzahl der generierten Paare verringert sich bei unserem Verfahren um eine durch die Anzahl und Grösse der Koreferenzmengen bestimmten Betrag. Je mehr NPs bei geringer Anzahl von Koreferenzmengen (aber grösser Null) in diesen Koreferenzmengen gebunden sind, desto weniger Paare werden generiert (siehe Abschnitt 'Experimente' für

konkrete Zahlen). Der Grund: ein Anapherkandidat m_i wird nur mit einem Element jeder Koreferenzmenge gepaart. Sei bei einem Fenster von 3 Sätzen die Kardinalität von $I = 10$ (also 10 NPen) und C bestehe aus einer einzigen Koreferenzmenge, die 6 Elemente aus I enthalte, dann wird m_{10} (die linear gesehen letzte zu integrierende NP) nur mit 5 NPen statt mit 9 gepaart: einem Element der Koreferenzmenge und den 3 gepufferten. Auf diese Weise reduziert sich auch die Anzahl der negativen Beispiele, da für jede Koreferenzmenge (egal, ob m_i dazu gehört oder nicht) ja immer nur ein Glied betrachtet wird.

Das inkrementelle Verfahren gibt uns neue, koreferenzmengebezogene Merkmale an die Hand. Wir können daher weitere, bislang in der Literatur nicht verwendete Merkmale definieren:

- stammt der Antezedenzkandidat aus dem Puffer oder einer Koreferenzmenge?
- Anzahl der gefundenen Kandidaten
- Anzahl der momentanen Koreferenzmengen
- Neueröffnung einer Koreferenzmenge oder Erweiterung einer bestehenden?

(die folgenden Features beziehen sich auf die ausgewählte Koreferenzmenge)

- wieviele Nomen hat die Koreferenzmenge?
- Kardinalität der Koreferenzmenge

Unsere empirischen Ergebnisse zeigen, dass der Nutzen dieser Merkmale vom Anapherntyp abhängt.

4 Vorverarbeitung

Die Vorverarbeitung dient der Extraktion linguistischer (morphologischer, syntaktischer und semantischer) Beschreibungen, die beim Filtern von Paarkandidaten bzw. als Merkmale beim Machine Learning verwendet werden. Wir verwenden Gertwol (Lingsoft, 1994), den TreeTagger (Schmid, 1994), GermaNet (Hamp & Feldweg, 1997), Wortschatz Leipzig (<http://www.wortschatz.uni-leipzig.de>), Wikipedia und den Parser Pro3GresDe (Sennrich et al., 2009).

Neben der Bestimmung des Lemmas und der morphologischen Kategorien, führt das Morphologianalysetool Gertwol auch eine Nomendekomposition durch, was sehr hilfreich ist, da Komposita oft nicht in GermaNet gefunden werden,

jedoch nach der Dekomposition ihre semantische Klasse anhand des Kompositakopfes oft richtig bestimmt werden kann.

Numerus, Genus und Person sind wesentlich für das Ausfiltern von sicheren negativen Paaren. Es gibt jedoch das Problem der Unterspezifikation und Ambiguität, z.B. bei den Pronomen 'sie', 'sich' und 'ihr'.

Die Named-Entity Erkennung ist musterbasiert und benutzt eine grosse Liste von Vornamen (53'000), wobei das Geschlecht zum Vornamen bekannt ist. Wir haben zudem aus der deutschen Wikipedia alle Artikel, deren Suchterm ein Mehrwortterm ist, extrahiert (z.B. 'Berliner Sparkasse') und, falls verfügbar, die zugehörige Wikipediakategorie (und diese, falls möglich, auf GermaNet abgebildet). GermaNet bzw. Wortschatz Leipzig liefern Synonyme und Hyponyme.

Pro3GresDe, ein hybrider Dependenzparser für das Deutsche, der eine handgeschriebene Grammatik mit einer statistischen Komponente zur Disambiguierung kombiniert, liefert u.a. die grammatische Funktion von NPen.

5 Experimente

Die folgenden empirischen Ergebnisse beruhen auf der TüBa-D/Z, einer Baumbank, die ebenfalls mit Koreferenzannotationen versehen wurde (24'000 annotierte Sätze in unserer Version), vgl. (Naumann, 2006). Als Klassifizierer verwenden wir das ähnlichkeitsbasierte Lernverfahren TiMBL (Daelemans et al., 2004). Als Evaluationsmass wird nicht der MUC-Scorer verwendet, sondern der ECM bzw. CEAf aus (Luo, 2005). Beim CEAf erfolgt zuerst eine Alinierung von Koreferenzmengen des Gold Standard mit den vom System gefundenen. Präzision ist dann die Anzahl der richtigen Elemente pro alinierter gefundener Menge durch die Anzahl der gefundenen, bei der Ausbeute wird entsprechend durch die Anzahl der tatsächlichen Elemente der Gold Standard Menge geteilt. Der CEAf ist ein strenges Mass, da u.U. auch in nicht alinierbaren Mengen richtige Paare existieren. Der Vorteil: der CEAf ermittelt die Güte der Koreferenzmengenpartitionierung.

Unser Ansatz ist filterbasiert, d.h. Paare, die die Filter nicht passieren, werden als negativ klassifiziert. Darunter sind viele *false negatives* und zwar vor allem im Bereich der Nominalanaphern. Dies sei am Beispiel der ersten 5000 Sätze il-

	Ohne Filterung			Mit Filterung		
	F-Mass	Präzision	Ausbeute	F-Mass	Präzision	Ausbeute
Nomen	72.70	69.53	76.17	62.61	63.70	61.55
Personalpronomen	60.42	62.05	58.88	58.86	60.64	57.19
Relativpronomen	56.25	57.91	54.68	55.97	57.65	54.39
Possessivpronomen	56.06	57.35	54.82	55.81	57.18	54.51
Reflexivpronomen	55.68	57.11	54.32	54.16	55.64	52.77
Gesamt	-	-	-	75.31	81.58	69.95
System	-	-	-	53.86	54.64	53.09

Figure 2: CEAF-Werte bei perfekten Einzelklassifikatoren des inkrementellen Systems für die ersten 5000 Sätze (in %). Wie gut wäre das System, wenn es einzelne Wortklassen perfekt auflösen würde? Mit Filterung heisst, dass nur die Paare perfekt ausgelöst werden, die entsprechende Filter passieren. Ohne Filterung bedeutet, dass alle gemäss Gold Standard positiven Paare der jeweiligen Wortklasse perfekt aufgelöst werden (die anderen Wortklassen werden vom realen System verarbeitet). Der Unterschied zwischen mit und ohne Filterung bezeichnet die Güte der Filter pro Wortklasse, die Unterschiede zum System, wie sehr die imperfekte Auflösung der Wortklasse das System drückt.

Verfahren	F-Mass	Präzision	Ausbeute
Nicht-inkrementell, ein Klassifizierer	44.04	55.60	36.48
Nicht-inkrementell, mehrere Klassifizierer	49.35	53.67	45.69
Inkrementell, ein Klassifizierer	50.66	52.54	48.93
Inkrementell, mehrere Klassifizierer	52.79	52.88	52.70
Salienz	51.41	52.03	50.82

Figure 3: CEAF-Werte der fünfachen Kreuzvalidierung (in %)

lustriert. In Abbildung (Figure) 2 werden die oberen Schranken (*upper bound*) mit und ohne Filter aufgelistet. Die tatsächliche Performanz (im gewählten Fold) des inkrementellen Systems mit mehreren Klassifizierern ist: F-Mass = 53.86%, Präzision = 54.64% Ausbeute = 53.09% Hätten wir z.B. eine perfekte Nominalanaphernresolution (die erste Zeile), dann könnten das System unter sonst unveränderten Bedingungen (die anderen Anapherarten werden weiterhin vom System aufgelöst), maximal 62.61% F-Mass erreichen (mit Filter); ohne Filter wären es 72.70%. Dies zeigt zweierlei. Nominalanaphern sind tatsächlich das Problem, wie die 9% Differenz zwischen Systemwert (53.86%) und perfektem Wert (62.61%) zeigt. Daneben erklärt die abermalige 10% Differenz zur perfekten Auflösung ohne Filter die insgesamt schlechte Performanz von 53.86%: insgesamt 19% Performanzverlust durch die Nominalanaphern. Im Vergleich zu den Reflexivpronomen. Hier ist die Differenz nur 0.3% (54.16% - 53.86%) zur perfekten Auflösung mit Filter und nur 1.8% zur perfekten Auflösung ohne Filter (55.68% - 53.86%).

Wir beschreiben nun unsere Experimente (fünffach kreuzvalidiert) zur Bestimmung des besten Ansatzes zur Koreferenzresolution, vgl. Abbildung (Figure) 3. Als Baseline dient ein nicht-inkrementelles Verfahren, das, wie alle Varianten, bzgl. Merkmalauswahl (features des Klassifizierers) optimiert wurde. Wir unterscheiden zwischen der Verwendung von einem und mehreren POS-spezifischen Klassifizierern.

Für das nicht-inkrementelle Verfahren mit nur einem Klassifizierer hat sich folgende Merkmalsmenge als am performantesten erwiesen: Salienz der NPen, grammatische Funktionen der NPen und ob diese parallel sind, Wortklasse der Köpfe der NPen und eine separate Kodierung der POS-Kombination, semantische Kompatibilität der NPen, Die Häufigkeit der NPen im Segment, ob der Antezedenskandidat der nächste kompatible zur Anapher ist, ob sich die NPen in direkter Rede befinden².

²Dass hier Distanz nicht verwendet wird, liegt daran, dass die Distanzmerkmale sich bei der Auflösung nominaler Anaphern als schlecht erwiesen haben. Da relativ viele nominale Paare bewertet werden müssen, schadet Distanz einem Einzelklassifizierer, der alle Arten von Anaphern bewerten

Dieser Ansatz (mit einem F-Mass von 44.04%) dient als erste Baseline. Die Verwendung mehrerer Klassifizierer (zweite Baseline) bringt eine Verbesserung um über 5% F-Mass (auf 49.35%). Das liegt daran, dass die Merkmale unterschiedlichen Einfluss auf die POS-spezifischen Klassifizierer haben. Distanz kann z.B. für Pronominalanaphern eingesetzt werden, bei den Nominalanaphern bringt sie nichts. Folgende Merkmalsmengen haben sich als am effektivsten herausgestellt:

- Nominalanaphern: Häufigkeiten der NPen im Segment, grammatische Funktionen der NPen, semantische Kompatibilität, Definitheit des Antezedenzkandidaten, ob sich der Anaphernkandidat in direkter Rede befindet und ein Indikator für den Filter, der das Paar generiert hat (Stringmatch, GermaNet oder Wortschatz Leipzig).
- Personalpronomen: Distanz in Sätzen und Markables, Salienz der NPen, Einbettungstiefe der NPen, Wortklasse der Köpfe der NPen.
- Relativpronomen: Distanz in Markables, Salienz der NPen, grammatische Funktion der NPen, Einbettungstiefe des Anaphernkandidaten, ob der Antezedenzkandidat der nächste kompatible zur Anapher ist.
- Reflexivpronomen: Distanz in Markables, Salienz des Antezedenzkandidaten, Wortklasse der Köpfe der NPen und Kodierung der POS-Kombination, grammatische Funktionen, Einbettungstiefe der NPen, ob der Antezedenzkandidat der nächste kompatible zur Anapher ist.
- Possessivpronomen: Salienz der NPen, Distanz in Sätzen, Einbettungstiefe der NPen, grammatische Funktionen der NPen und ob diese parallel sind, Wortklasse des Kopfs des Anaphernkandidaten, ob der Antezedenzkandidat der nächste kompatible zur Anapher ist.

Wie oben erwähnt wurde, können im inkrementellen Modell Merkmale definiert werden, die sich auf die (entstehenden) Koreferenzmengen beziehen. Insgesamt hat sich die Verwendung dieser Merkmale als ambivalent herausgestellt:

muss.

Nicht für alle Klassifizierer sind sie hilfreich. Bei der Verwendung nur eines Klassifizierers werden im inkrementellen Modell drei der erwähnten Merkmale verwendet: Kardinalität der Koreferenzmenge, Anzahl der Nomen in der Koreferenzmenge, ob eine neue Koreferenzmenge eröffnet wird. Ansonsten werden die gleichen Merkmale wie im nicht-inkrementellen Modell verwendet. Der Unterschied zu Baseline 2 ist mit 1.3% gering, doch spürbar.

Bei der Verwendung mehrerer Klassifizierer haben die Koreferenzmengen bezogenen Merkmale nur einen signifikanten Einfluss auf die Klassifizierer der Personal- und Possessivpronomen. Die Anzahl vorhandener Koreferenzmengen wird bei beiden verwendet, die Kardinalität der Koreferenzmenge zusätzlich bei den Personalpronomen. Ansonsten werden auch hier die gleichen Merkmale wie im nicht-inkrementellen Verfahren verwendet.

Bei der Verwendung mehrerer Klassifizierer werden, gegenüber Baseline 2, fast drei Prozentpunkte F-Mass dazugewonnen (52.79% vgl. mit 49.35%). Die Verbesserungen, die durch die Verwendung mehrerer Klassifizierer erreicht werden, entsteht v.a. durch einen Anstieg der Ausbeute. Auffallend ist, dass das Verhältnis von Präzision zu Ausbeute im inkrementellen Modell ausgeglichener ist als im nicht-inkrementellen.

Bezüglich Laufzeit ist festzuhalten, dass die nicht-inkrementellen Verfahren nicht nur mehr negative, sondern auch mehr positive Instanzen generieren, da alle Mentions aus den Koreferenzmengen für die Paargenerierung zugänglich sind. Diese zusätzlichen positiven und negativen Instanzen erhöhen die Laufzeit beträchtlich. Im letzten Fold (etwa 5000 Sätze) der Kreuzvalidierung z.B. generiert das nicht-inkrementelle Modell für das Training 23024 positive und 109344 negative Instanzen. Das inkrementelle Modell hingegen erstellt nur 10957 positive und 76955 negative Paare. Das entspricht bei den positiven Instanzen einer Reduktion von über der Hälfte, bei den negativen Instanzen um rund 30%. Da alle Verfahren die gleichen Filter verwenden, gehen in den inkrementellen Ansätzen keine *true mentions* verloren. Die Reduktion entsteht dadurch, dass Paare nur mit einem Element der jeweiligen Koreferenzmengen gebildet werden. Auch die Client-Server-Architektur des inkrementellen Modells beschleunigt die Laufzeit, da die TiMBL-

Klassifizierer nicht für jede Klassifikation neu gestartet werden müssen.

Die letzte Zeile von Abbildung 3 gibt das Resultat der rein salienzbasierten Variante des inkrementellen Ansatzes wieder. Es schneidet erstaunlich gut ab und liegt mit 51.41% um 1.4% unter der Bestmarke. Diese gute Performanz bei der Einfachheit der Implementierung und der im Vergleich enorm kurzen Laufzeit, sind gute Argumente gegen die aufwändigeren Implementationen von Machine Learning Ansätzen. Dazu kommt, dass die Optimierung von Merkmalsmengen in Machine Learning Ansätzen einerseits nötig, andererseits aber auch zeitintensiv und die Auswirkungen einzelner Merkmalsetzungen unvorhersehbar ist. Die erzielten Verbesserungen aufgrund von Mutationen der Merkmalsmengen können außerdem linguistisch oft nur schwer begründet werden, resp. entziehen sich der Intuition. Ein Argument für die Verwendung von ML Verfahren ist aber die Behandlung von Bridging Anaphern, die in unserem salienzbasierten Verfahren nicht aufgelöst werden.

6 Literaturdiskussion

Die Arbeit von (Soon et al., 2001) ist ein prototypisches, oft reimplementiertes (Baseline-)Modell zur Anaphernresolution, das auf paarweiser Klassifikation und statistischen Verfahren basiert.

Eines der wenigen inkrementellen Modelle ist (Yang et al., 2004). Im Gegensatz zum vorliegenden Modell gibt es in diesem Ansatz für's Englische jedoch nur ein einziges echtes koreferenzmengenbezogenes Merkmal: die Anzahl der Elemente einer Koreferenzmenge.

Es gibt einige wenige Arbeiten zur Koreferenzresolution für das Deutsche, die meisten nutzen die Koreferenzannotation der Baumbank TüBa-D/Z. Uns ist kein System bekannt, das basierend auf realen Vorverarbeitungskomponenten sowohl Pronominal- als auch Nominalanaphernresolution modelliert. Die sehr aufschlussreiche Untersuchung von (Schiehlen, 2004) ist ebenfalls auf Pronominalanaphern beschränkt, zeigt aber wie tief die empirischen Werte tatsächlich liegen, wenn man reale Komponenten verwendet statt einer Baumbank.

(Versley, 2006) hat – auf der Basis einer Teilmenge der TüBa-D/Z – zahlreiche Experimente zur Nominalanaphernresolution durchgeführt (z.B. verschiedene statistische Masse um z.B.

Selektionsrestriktionen zu modellieren). Sein Befund, dass wenn Information aus GermaNet verfügbar ist, diese dann statischer Information überlegen ist, hat uns dazu inspiriert, GermaNet durch Wikipedia und Wortschatz Leipzig zu komplementieren und auf statistische Berechnungen zu verzichten.

Neben GermaNet und einem pattern-basierten Ansatz, verwenden (Goecke et al., 2008) Latent Semantic Indexing bei der Nominalanapherauflösung. Die empirische Analyse erfolgt anhand eines kleinen, von den Autoren eigens annotierten Korpus.

Modelle für Pronominalanaphern werden in einer Reihe von Arbeiten aus Tübingen diskutiert. Die empirischen Ergebnisse basieren auf Goldstandardinformation, so wird z.B. in (Wunsch et al., 2009) eine perfekte Morphologie und funktionale Information der TüBa-D/Z Baumbank verwendet. Diese Arbeit versucht das Problem der Übergenerierung von negativen Beispielen durch Sampling zu lösen. Das vorliegende inkrementelle Modell kann als Alternative dazu aufgefasst werden. Die Reduktion von Trainingsinstanzen ist ein natürlicher Nebeneffekt unseres inkrementellen Verfahrens.

7 Zusammenfassung und Ausblick

Es wurde ein Verfahren zur Koreferenzresolution für das Deutsche vorgestellt, das von realen Verarbeitungsmodalitäten ausgeht und sowohl Pronominal- als auch Nominalanaphern behandelt. Wir können festhalten, dass ein filterbasiertes inkrementelles Verfahren auf der Basis anaphernspezifischer Klassifizierer am besten arbeitet. Überraschenderweise ist der Abstand zu einem einfachen salienzbasierten System gering.

Die empirischen Werte sind mit knapp 52.79% F-Mass nicht berauschend. Schuld daran sind Fehler in den Annotationen der TüBa-D/Z (fehlende Annotationen bei Pronomen und matchenden Named Entities), Fehler beim Vorverarbeiten (z.B. Morphologie) und die Unterspezifikation im Bereich der Nominalanaphern (z.B. GermaNet-Lücken). Nominalanaphern bleiben die grosse Herausforderung.

Unser inkrementelles Verfahren ermöglicht eine natürliche Reduktion zu lernender Beispiele beim Vektorgenerieren, es ermöglicht uns darüberhinaus die Verwendung neuer Features wie z.B. die Anzahl der Koreferenzmengen. Nicht

alle neuen Merkmalen helfen die Empirie zu verbessern, und unterschiedliche Anapherntypen profitieren von unterschiedlichen Merkmalen.

Unser Modell ist noch nicht ausgeschöpft. Verbesserungen erwarten wir u.a. im Bereich der Nominalanaphern. In jedem Fall aber liefert unser System eine nicht geschönte Baseline für die Ko-referenzresolution des Deutschen.

Danksagung

Die Arbeiten zu unserem Projekt werden vom Schweizerischen Nationalfonds unterstützt (Nummer 105211-118108).

References

- Egon Balas. 1965. An Additive Algorithm for Solving Linear Programs with Zero-one Variables. *Operations Research*, 13(4).
- Walter Daelemans and J. Zavrel and K. van der Sloot and A. van den Bosch. 2004. *TiMBL: Tilburg Memory-Based Learner*. Techn. Report. Tilburg University.
- Niyu Ge and John Hale and Eugene Charniak. 1998. A Statistical Approach to Anaphora Resolution. *Proc. of the Sixth Workshop on Very Large Corpora*.
- Daniela Goecke, Maik Stührenberg, Tonio Wandmacher. 2008. A Hybrid Approach to Resolve Nominal Anaphora. *LDV Forum*, 1(23).
- Birgit Hamp and Helmut Feldweg. 1997. GermaNet—a Lexical-Semantic Net for German. *Proc. of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.
- Erhard W. Hinrichs and Katja Filippova and Holger Wunsch. 2005. A Data-driven Approach to Pronominal Anaphora Resolution in German. *Proc. of RANLP*.
- Manfred Klenner. 2007. Enforcing Consistency on Coreference Sets. *Proc. of the Ranlp*.
- Manfred Klenner and Étienne Ailloud. 2009. Optimization in Coreference Resolution Is Not Needed: A Nearly-Optimal Zero-One ILP Algorithm with Intensional Constraints. *Proc. of the EACL*.
- Shalom Lappin and Herbert J. Leass. 1994. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*.
- Lingsoft. 1994. Gertwol. Questionnaire for Morpholympics. *LDV-Forum*, 11(1).
- Xiaoqiang Luo. 2005. On Coreference Resolution Performance Metrics. *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*.
- Ruslan Mitkov. 1998. Robust Pronoun Resolution with Limited Knowledge. *Proc. of the ACL*. Montreal, Quebec, Canada.
- Karin Naumann. 2006. *Manual for the Annotation of Indocument Referential Relations*. Tech. Report, Universität Tübingen.
- Michael Schiehlen. 2004. Optimizing Algorithms for Pronoun Resolution. *Proc. of the 20th International Conference on Computational Linguistics*.
- Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proc. of the Conference on New Methods in Language Processing*.
- Rico Sennrich and Gerold Schneider and Martin Volk and Martin Warin. 2009. A New Hybrid Dependency Parser for German. *Proc. of the German Society for Computational Linguistics and Language Technology 2009 (GSCL 2009)*. Potsdam.
- Wee Meng Soon and Hwee Tou Ng and Daniel Chung Young Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*.
- Holger Wunsch and Sandra Kübler and Rachael Cantrell. 2009. Instance Sampling Methods for Pronoun Resolution. *Proc. of RANLP*. Borovets.
- Yannick Versley. 2006. A Constraint-based Approach to Noun Phrase Coreference Resolution in German Newspaper Text. *Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS)*.
- Xiaofeng Yang and Jian Su and Guodong Zhou and Chew Lim Tan. 2004. An NP-Cluster Based Approach to Coreference Resolution. *Proc. of Coling*.

Determining the Degree of Compositionality of German Particle Verbs by Clustering Approaches

Natalie Kühner and Sabine Schulte im Walde

Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart

Germany

{kuehnene, schulte}@ims.uni-stuttgart.de

Abstract

This work determines the degree of compositionality of German particle verbs by two soft clustering approaches. We assume that the more compositional a particle verb is, the more often it appears in the same cluster with its base verb, after applying a probability threshold to establish cluster membership. As German particle verbs are difficult to approach automatically at the syntax-semantics interface, because they typically change the subcategorisation behaviour in comparison to their base verbs, we explore the clustering approaches not only with respect to technical parameters such as the number of clusters, the number of iterations, etc. but in addition focus on the choice of features to describe the particle verbs.

1 Introduction

A multi-word expression (MWE) is a combination of two or more simplex words,¹ covering compounds as well as collocations. From a semantic point of view, multi-word expressions are either considered as idiosyncratic (Sag et al., 2002; Villavicencio et al., 2005; Fazly and Stevenson, 2008), i.e., non-compositional, or alternatively the MWE compositionality is assumed to be on a continuum between entirely compositional/transparent and entirely non-compositional/opaque expressions. We conform to the latter view, and consider multi-word expressions as a composition of simplex words which may or may not be entirely predictable on the basis of standard rules and lexica. This view is in line with recent work on multi-word expressions,

e.g., McCarthy et al. (2003; 2007), and also theoretical considerations about compositionality, cf. Kavka (2009).

Addressing the compositionality of multi-word expressions is a crucial ingredient for lexicography (concerning the question of whether to lexicalise a MWE) and Natural Language Processing applications (to know whether the expression should be treated as a whole, or through its parts, and what the expression means). We are interested in determining the degree of compositionality of one empirically challenging class of German multi-word expressions, i.e., German particle verbs, productive compositions of a base verb and a prefix particle. The work relies on a Studienarbeit by the first author (Kühner, 2010).

We propose two clustering approaches to address the compositionality of particle verbs. The core idea is that the compositionality of the multi-word expressions is determined by the co-occurrence of the particle verbs and the respective base verbs within the same clusters. I.e., we assume that the more compositional a particle verb is, the more often it appears in the same cluster with its base verb. Note that our idea restricts the compositionality of multi-word expressions to the relationship between particle and base verb and thus for the time being ignores the contribution of the particle. As we are relying on soft clustering approaches, cluster membership is represented by a probability. We transfer the probabilistic membership into a binary membership by establishing a membership cut-off, i.e., only verbs above a certain probability threshold are considered to be cluster members.

German particle verbs are an empirical challenge because they are difficult to approach automatically at the syntax-semantics interface: they

¹Note that the definition of multi-words is not straightforward or agreed upon Lieber and Stekauer (2009a). Our definition is one possibility among many, but has generally been adopted by computational linguistics.

typically change the subcategorisation behaviour in comparison to their base verbs, cf. Section 2. Consequently, we explore the clustering approaches not only with respect to technical parameters such as the number of clusters, the number of iterations, etc. but in addition focus on the choice of features to describe the particle verbs. The compositionality scores as predicted by the clustering approaches are evaluated by comparison against human judgements, using the Spearman rank-order correlation coefficient.

The remainder of the paper is organised as follows. Section 2 introduces the reader into German particle verbs. Following an overview of the clustering approaches in Section 3, we then describe the experiments (Section 4) and the results (Section 5).

2 German Particle Verbs

German particle verbs (PVs) are productive compositions of a base verb (BV) and a prefix particle, whose part-of-speech varies between open-class nouns, adjectives, and verbs, and closed-class prepositions and adverbs. This work focuses on preposition particles.

Particle verb senses are assumed to be on a continuum between transparent (i.e. compositional) and opaque (i.e. non-compositional) with respect to their base verbs. For example, *abholen* ‘fetch’ is rather transparent with respect to its base verb *holen* ‘fetch’, *anfangen* ‘begin’ is quite opaque with respect to *fangen* ‘catch’, and *einsetzen* has both transparent (e.g. ‘insert’) and opaque (e.g. ‘begin’) verb senses with respect to *setzen* ‘put/sit (down)’. Even though German particle verbs constitute a significant part of the verb lexicon, most work is devoted to theoretical investigations, such as (Stiebels, 1996; Lüdeling, 2001; Dehé et al., 2002). To our knowledge, so far only (Aldinger, 2004; Schulte im Walde, 2004; Schulte im Walde, 2005; Rehbein and van Genabith, 2006; Hartmann et al., 2008) have addressed German particle verbs from a corpus-based perspective.

This work addresses the degrees of compositionality of preposition particle verbs by clustering and then comparing the cluster memberships of the particle and base verbs. Clustering particle verbs and base verbs in turn requires the definition of empirical properties. This work relies on an automatic induction of distributional features from large-scale German corpus data, cf. Sec-

tion 4.1. While inducing the distributional information is not difficult per se, German particle verbs face an empirical challenge: In general, subcategorisation properties are a powerful indicator of verb semantic relatedness and could thus point us towards the strength of relatedness between particle and base verbs (Dorr and Jones, 1996; Schulte im Walde, 2000; Korhonen et al., 2003; Schulte im Walde, 2006; Joannis et al., 2008, among others) because distributional similarity with respect to subcategorisation frames (even by themselves) corresponds to a large extent to semantic relatedness. German particle verbs are difficult to approach automatically at the syntax-semantics interface, however, because they typically change the subcategorisation behaviour in comparison to their base verbs. For example, even though *anlächeln* in example (1)² taken from Lüdeling (2001) is strongly compositional, its subcategorisation properties differ from those of its base verb; thus, automatic means that rely on subcategorisation cues might not recognise that *anlächeln* is semantically related to its base verb. Theoretical investigations (Stiebels, 1996) as well as corpus-based work (Aldinger, 2004) have demonstrated that such changes are quite regular, independent of whether a particle verb sense is compositional or not.

- (1) Sie lächelt.
‘She smiles.’
*Sie lächelt [NP_{acc} ihre Mutter].
‘She smiles her mother.’
Sie lächelt [NP_{acc} ihre Mutter] an.
‘She smiles her mother at.’

We believe that there are basically two strategies to address the empirically challenging class of multi-word expression from a semantic perspective: (i) avoid subcategorisation-based distributional features at the syntax-semantics interface, or (ii) incorporate the syntax-semantics subcategorisation transfer into the distributional information, cf. (Aldinger, 2004; Hartmann et al., 2008). This paper adheres to strategy (i) and basically excludes the notion of syntax from the distributional descriptions. For comparison reasons, we include an experiment that incorporates syntactic functions.

²Note that German particle verbs are separable, in contrast to the class of German prefix verbs that share many properties with the class of particle verbs but are inseparable (among other differences).

3 Clustering Approaches: LSC and PAC

Two soft clustering approaches were chosen to model the compositionality of German particle verbs, Latent Semantic Classes (LSC) and Predicate-Argument Clustering (PAC). Using soft clustering, each clustering object (i.e., the particle and base verbs) is assigned to each cluster with a probability between 0 and 1, and all probabilities for a certain verb over all clusters sum to 1. Cluster membership is then determined according to a probability threshold, cf. Section 4.2. In the following, we introduce the two clustering approaches.

3.1 Latent Semantic Classes

The Latent Semantic Class (LSC) approach is an instance of the Expectation-Maximisation (EM) algorithm (Baum, 1972) for unsupervised training on unannotated data, originally suggested by Mats Rooth (Rooth, 1998; Rooth et al., 1999). We use an implementation by Helmut Schmid. LSC cluster analyses define two-dimensional soft clusters which are able to generalise over hidden data. They model the selectional dependencies between two sets of words participating in a grammatical relationship. LSC training learns three probability distributions, one for the probabilities of the clusters, and a joint probability distribution for each lexical class participating in the grammatical relationship, with respect to cluster membership, thus the two dimensions. In our case, one dimension are the verbs (particle and base verbs), and one dimension are corpus-based features. Equation (2) provides the probability model for verb-feature pairs (v and f , respectively). Note that in our case the second dimension is crucial for the cluster analysis, but for determining the compositionality of the particle verbs, we only consider the cluster probabilities of dimension one, i.e., the particle and base verbs. Table 1 presents an example cluster that illustrates the verb and the feature dimensions, presenting the most probable verbs and direct object nouns within the cluster. The cluster is a nice example of compositional particle verbs (*verschicken*, *abschicken*, *zuschicken*) clustered together with their base verb (*schicken*).

$$\begin{aligned} p(v, f) &= \sum_{c \in \text{cluster}} p(c, v, f) \\ &= \sum_{c \in \text{cluster}} p(c) p(v|c) p(f|c) \end{aligned} \quad (2)$$

3.2 Predicate-Argument Clustering

Predicate-Argument Clustering (PAC) is an extension of the LSC approach that explicitly incorporates selectional preferences (Schulte im Walde et al., 2008). The PAC model provides a combination of the EM algorithm and the Minimum Description Length (MDL) principle (Rissanen, 1978), and refines the second dimension by explicit generalisations based on WordNet (Fellbaum, 1998) and the MDL principle. For example, instead of high probabilities of the nouns *Milch* ‘milk’, *Kaffee* ‘coffee’, *Tee* ‘tea’ within dimension two of a cluster, PAC might identify the generalising WordNet concept *Getränk* ‘beverage’. Note that with PAC the second dimension only makes sense if WordNet provides useful generalisation information concerning that dimension, which effectively restricts the word class of the second dimension to nouns.

The PAC model is estimated through the joint probability of a verb v , a subcategorisation frame type f , and the complement realisations n_1, \dots, n_k , cf. Equation (3). In addition to the LSC parameters in Equation (2), $p(r|c, f, i)$ is the probability that the i th complement of frame f in cluster c is realised by WordNet (*wn*) concept r , and $p(n|r)$ is the probability that the WordNet concept r is realised by the complement head n . Table 2 presents an example cluster where dimension two is a generalisation of WordNet concepts over PP arguments. Dimension one contains the most probable verbs in the cluster; dimension two is a selection of the most probable concepts from different hierarchical levels, plus example instances. As we are working on German data, we use the German Wordnet, i.e., *GermaNet* (Kunze, 2000).

$$p(v, f, n_1, \dots, n_k) = \sum_c p(c) p(v|c) p(f|c) * \prod_{i=1}^k \sum_{r \in \text{wn}} p(r|c, f, i) p(n_i|r) \quad (3)$$

4 Clustering Experiments

To setup the clustering experiments, we need to specify the linguistic parameters (i.e., the choice of verbs and features), and the technical parameters, cf. Sections 4.1 and 4.2, respectively. The evaluation is described in Section 4.3.

4.1 Data

As corpus data basis, we relied on approx. 560 million words from the German web corpus

<i>dimension 1: verbs</i>		<i>dimension 2: direct object nouns</i>	
schicken	‘send’	Artikel	‘article’
verschicken	‘send’	Nachricht	‘message’
versenden	‘send’	E-Mail	‘email’
nachweisen	‘prove’	Brief	‘letter’
überbringen	‘deliver’	Kind	‘child’
abonnieren	‘subscribe to’	Kommentar	‘comment’
zusenden	‘send’	Newsletter	‘newsletter’
downloaden	‘download’	Bild	‘picture’
bescheinigen	‘attest’	Gruß	‘greeting’
zustellen	‘send’	Soldat	‘soldier’
abschicken	‘send off’	Foto	‘photo’
zuschicken	‘send’	Information	‘information’

Table 1: Example LSC cluster.

<i>dimension 1: verbs</i>		<i>dimension 2: WN concepts over PP arguments</i>	
steigen	‘increase’	Maßeinheit	‘measuring unit’
zurückgehen	‘decrease’	e.g., Jahresende	‘end of year’
geben	‘give’	Geldeinheit	‘monetary unit’
rechnen	‘calculate’	e.g., Euro	‘Euro’
wachsen	‘grow’	Transportmittel	‘means of transportation’
ansteigen	‘increase’	e.g., Fahrzeug	‘automobile’
belaufen	‘amount to’	Gebäudeteil	‘part of building’
gehen	‘go’	e.g., Dach	‘roof’
zulegen	‘add’	materieller Besitz	‘material property’
anheben	‘increase’	e.g., Haushalt	‘budget’
kürzen	‘reduce’	Besitzwechsel	‘transfer of property’
stehen	‘stagnate’	e.g., Zuschuss	‘subsidy’

Table 2: Example PAC cluster.

deWaC (Baroni and Kilgarriff, 2006), after the corpus was preprocessed by the Tree Tagger (Schmid, 1994) and by a dependency parser (Schiehlen, 2003). The corpus portion contains more than 50,000 verb types (from verb-first, verb-second and verb-final clauses), which we restricted to those with a frequency above 1,000 and below 10,000, to avoid very low and very high frequent types, as they notoriously produce noise in clustering. In addition, we made sure that all verbs needed in the evaluation were covered, ending up with 2,152 verb types (comprising both particle and base verbs). The latter step, however, included some low and high frequent verbs, as many particle verbs are low frequent, and many base verbs are high frequent.

Concerning the feature choice, we relied on the main verb argument types, covering subjects, direct objects and pp objects. I.e., we used as in-

put verb–noun pairs where the nouns were (a) subjects, or (b) objects, or (c) pp objects of the verbs. We used the information separately and also (d) merged without reference to the syntactic function, as we largely ignored syntax. The underlying assumption for this rather crude simplification refers to the observation that the selectional preferences of particle verbs overlap with those of semantically similar verbs, but not necessarily in identical syntactic functions, cf. Schulte im Walde (2004). In comparison to (d), we (e) merged the pairs, while keeping the reference to the syntactic functions. The feature choice –more specifically: comparing (d) with (e)– is based on that in Schulte im Walde (2005). We wanted to compare the individual argument types with respect to their potential in addressing particle verb compositionality despite the syntax transfer hurdle. As direct objects and pp objects often re-

main the same function after the syntax-semantics particle–base transfer, they were supposed to provide more interesting results than subjects, which often fulfil more general roles. In addition, the syntax-unmarked input was supposed to provide better results than the syntax-marked input, because of the syntax transfer hurdle. The input variants are referred to as (a) *subj*, (b) *obj*, (c) *pp*, (d) *n-syntax*, and (e) *n+syntax*. Table 3 lists the number of input tokens and types according to the feature choices.

input	tokens	types
subj	2,316,757	368,667
obj	3,532,572	446,947
pp	4,144,588	706,377
n+syntax	9,993,917	1,346,093
n-syntax	9,993,917	1,036,282

Table 3: Input data.

4.2 Method

The data were used for both LSC and PAC, with minor formatting differences. There are basically two input dimensions (verb and argument head) as described in Section 3. When including the function markers, they were added to the (second) noun dimension, e.g., *anfangen-Job* ‘begin-job’ would become *anfangen-obj:Job*.

As we wanted to explore the clustering potential with respect to various parameters, we varied the number of clusters: 20, 50, 100, and 200. In addition, we varied the probability to determine cluster membership: 0.01, 0.001, 0.0005, and 0.0001, thus directly influencing precision and recall, as higher probability thresholds include less verbs per cluster. All cluster analyses were trained over 200 iterations for LSC and 100 iterations for PAC, evaluating the results after 50, 100 (and 200) iterations.

4.3 Evaluation

For the evaluation of the experiments, we relied on a gold standard created by Hartmann (2008). She had collected compositionality judgements for 99 German particle verbs across 11 different preposition particles, and across 8 frequency bands (5, 10, 18, 30, 55, 110, 300, 10,000) plus one manually chosen verb per particle (to make sure that interesting ambiguous verbs were included). The frequency bands had been determined such that there

were approximately equally many particle verbs in each range.

Four independent judges had rated the compositionality of the 99 particle verbs between 1 (*completely opaque*) and 10 (*completely compositional*). The inter-rater agreement was significantly high ($W = 0.7548, \chi^2 = 274.65, df = 91, \alpha = 0.000001$), according to Kendall’s coefficient of concordance. The average ratings of the judges per particle verb are considered as the gold standard scores for our experiments. Table 4 presents a selection of the average scores for particle verbs with different degrees of compositionality. Note that there are ambiguous particle verbs, whose scores are the average values of the compositionality scores for the different meanings.

particle verb	score	
nachdrucken	‘reprint’	9.250
aufhängen	‘hang up’	8.500
ausschneiden	‘cut out’	8.250
vorgehen	‘go ahead’	6.875
	‘approach’	
abwaschen	‘do the dishes’	6.500
abschließen	‘close’	6.000
	‘finalise’	
nachweisen	‘prove’	5.000
anklagen	‘accuse’	4.500
zutrauen	‘feel confident’	3.250
umbringen	‘kill’	1.625

Table 4: Gold standard judgements.

The evaluation itself was performed as follows. For each cluster analysis and each probability threshold t , we calculated for each particle verb from the gold standard the proportion of how often it appeared in a cluster together with its base verb, in relation to the total number of appearances, cf. Equation (4). The ranked list of the cluster-based compositionality judgements was then compared against the ranked list of gold standard judgements, according to the Spearman rank-order correlation coefficient. This correlation is a non-parametric statistical test that measures the association between two variables that are ranked in two ordered series.

$$comp_{pv} = \frac{\sum_c p(pv, c) \geq t \wedge p(bv, c) \geq t}{\sum_c p(pv, c) \geq t} \quad (4)$$

The collection of the gold standard and the evaluation procedure were performed according to a comparable evaluation task for English particle verb compositionality in McCarthy et al. (2003). The parametric tests are described in Siegel and Castellan (1988).

5 Results

The correlation scores differ substantially according to the linguistic features and the parameters of the cluster analyses. Furthermore, the probability threshold that determined cluster membership directly influenced the number of particle verbs that were included in the evaluation at all. We focus on presenting the overall best results per feature (group) in Tables 5 and 6 for LSC and PAC, respectively, and comment on the overall patterns. The tables show

- the Spearman rank-order correlation coefficient (*corr*),
- the coverage (*cov*), i.e., the proportion of gold standard verbs included in the evaluation after applying the probability threshold,
- the *f-score* (F_1) of the correlation and coverage values as usually applied to precision and recall; it indicates a compromise between the correlation and the coverage, cf. Equation (5),
- the number of clusters,
- the number of iterations, and
- the membership threshold

of the best results.

$$f\text{-score} = \frac{2 * corr * cov}{corr + cov} \quad (5)$$

5.1 Technical Parameters

The best results per feature (group) as listed in the tables are reached with different numbers of clusters (ranging from 20 to 200); with LSC, the best results are obtained after all (i.e., 200) training iterations; with PAC, the best results are obtained sometimes after 50, sometimes after 100 iterations. So in the tables (and in general), there is no clear tendency towards an optimal number of clusters with respect to our task; concerning the optimal number of training iterations, LSC seems to profit most from the largest possible number of iterations (so it might be worth testing even more training iterations than 200), and PAC does not seem to have a strong preference.

The optimal probability threshold for cluster membership is difficult to judge about, as that value strongly depends on a preference for correlation vs. coverage. The lower the threshold, the more particle verbs are included in the clusters, so the recall (coverage) increases while the precision (correlation) decreases. The tables list the best results according to the f-score, but if one wanted to use the cluster analyses within an application that incorporates particle verb compositionality values, one would have to determine a favour for precision vs. recall, to identify the appropriate threshold. The best correlation results with an acceptable coverage of 50-60% go up to .433 (LSC, obj), and .236 (PAC, n-syntax). In general, the coverage is approx. 10-30% for a threshold of 0.01, 30-60% for a threshold of 0.001, 40-70% for a threshold of 0.0005, and 50-80% for a threshold of 0.0001.

Overall, the best f-score values go up to .499 for LSC and .327 for PAC, and the PAC results are in general considerably below the LSC results. The lowest f-scores go down to zero for both clustering approaches, and sometimes even reach negative values, indicating a negative correlation. In sum, our methods reach moderate correlation values, and considering that we have worked with very simple distributional features that ignored other than some basic information at the syntax-semantics interface, we regard this a reasonable result. The dependency of the correlation scores on the clustering parameters, however, remains largely unclear.

5.2 Linguistic Parameters

Concerning the linguistic features in the clustering, the picture differs with respect to LSC vs. PAC. With LSC, direct object and pp object information is obviously valuable in comparing particle verbs with base verbs, despite the transfer at the syntax-semantics interface, while subject information is not very helpful, as expected. Comparing the unions of syntactic functions with the individual functions, LSC profits more from the individual functions, while PAC profits more from the unions. In both approaches, the unmarked *n-syntax* condition outperforms the marked *n+syntax* condition, as expected, but the difference is not impressive.

Comparing LSC and PAC, we can identify various reasons for why the PAC results are considerably below the LSC results: (i) the dependency

input	best result			analysis		membership threshold
	corr	cov	f-score	clusters	iter	
obj	.433	.59	.499	100	200	.0005
subj	.205	.76	.323	50	200	.0001
pp	.498	.40	.444	20	200	.0005
n+syntax	.303	.54	.388	50	200	.0005
n-syntax	.336	.56	.420	100	200	.001

Table 5: LSC results.

input	best result			analysis		membership threshold
	corr	cov	f-score	clusters	iter	
obj	.100	.53	.168	100	50	.0005
subj	.783	.05	.094	20	50	.01
pp	.275	.21	.238	200	100	.01
n+syntax	.213	.61	.316	20	100	.0001
n-syntax	.236	.53	.327	200	100	.001

Table 6: PAC results.

of selectional preferences on the subcategorisation frames that represents a strength of PAC, does not play an important role in our task (rather, the reference to syntactic functions is supposed to have a negative influence on the prediction of compositionality, cf. Section 2); (ii) the high frequency (base) verbs included in the training data have a negative impact on cluster composition, i.e., many clusters created by PAC are dominated by few high-frequency verbs, which is sub-optimal in general but in our case has the additional effect that many compositionality predictions are 1 because it is very likely that for a specific particle verb also the base verb is in the cluster; (iii) the generalising property of PAC that would have been expected to help with the sparse data of the lexical heads, does not improve the LSC results but rather makes them worse.

Tables 7 and 8 present compositionality scores from the best LSC and the best PAC cluster analyses (cf. Tables 5 and 6), and relates them to the gold standard (gs) scores repeated from Table 4. Furthermore, the number of clusters in which the particle verb (pv), the respective base verb (bv) and both appeared, is given. While the LSC system scores are of course not perfect, we can see that there is a clear tendency towards higher overlap scores in the top half of the table, in comparison to the bottom half, even though the number of clusters the particle verbs appear in differ strongly. The only particle verb that clearly is not able to

subcategorise a direct object (i.e., *vorgehen* in both of its senses) is also a clear outlier in the quality of predicting the compositionality. In comparison to the LSC results, the PAC system scores are obviously worse, the main reason being that the high frequency base verbs appear in many of the 200 clusters, especially *gehen* and *bringen*.

In sum, the optimal clustering setup to predict particle verb compositionality (with respect to the best results in the tables, but also in more general) seems to use LSC with direct object or pp object information. On the one hand, the preference for these functions is intuitive (as many particle verbs as well as their base verbs are transitive verbs, e.g., *anbauen* ‘build, attach’, *nachdrucken* ‘reprint’, *umbringen* ‘kill’), but on the other hand the gold standard also includes many intransitive particle verbs (e.g., *aufatmen* ‘breathe’, *durchstarten* ‘touch and go’, *überschäumen* ‘foam over’) where at least direct objects intuitively cannot help with a compositionality rating.

5.3 Comparison with Related Work

McCarthy et al. (2003) predicted the degree of compositionality of English particle verbs. Their work is probably most closely related to our approach, and we adapted their evaluation method. Their prediction relies on nearest neighbourhood, assuming that the neighbours of particle verbs should be similar to the neighbours of the respective base verbs. The definition of neighbourhood is based on Lin’s thesaurus (Lin, 1998), and vari-

particle verb		#clusters			score	
		pv	bv	both	gs	system
nachdrucken	‘reprint’	2	5	1	9.250	0.500
aufhängen	‘hang up’	4	18	4	8.500	1.000
ausschneiden	‘cut out’	5	3	3	8.250	0.600
vorgehen	‘go ahead’	5	18	1	6.875	0.200
	‘approach’					
abwaschen	‘do the dishes’	1	4	1	6.500	1.000
abschließen	‘close’	2	2	1	6.000	0.500
	‘finalise’					
nachweisen	‘prove’	16	20	5	5.000	0.313
anklagen	‘accuse’	5	8	1	4.500	0.200
zutrauen	‘feel confident’	12	4	1	3.250	0.083
umbringen	‘kill’	2	2	0	1.625	0.000

Table 7: LSC gold standard judgements and system scores.

particle verb		#clusters			score	
		pv	bv	both	gs	system
nachdrucken	‘reprint’	0	13	0	9.250	–
aufhängen	‘hang up’	3	66	3	8.500	1.000
ausschneiden	‘cut out’	3	10	3	8.250	1.000
vorgehen	‘go ahead’	47	194	47	6.875	1.000
	‘approach’					
abwaschen	‘do the dishes’	1	9	1	6.500	1.000
abschließen	‘close’	63	98	48	6.000	0.762
	‘finalise’					
nachweisen	‘prove’	66	56	24	5.000	0.364
anklagen	‘accuse’	11	35	5	4.500	0.455
zutrauen	‘feel confident’	7	7	0	3.250	0.000
umbringen	‘kill’	11	190	11	1.625	1.000

Table 8: PAC gold standard judgements and system scores.

ous statistical measures for distributional similarity. The best result they achieve is a Spearman rank correlation of 0.490, which is slightly but not considerably better than our best results.

Concerning the feature choice to describe and compare German particle verbs and their base verbs (more specifically: comparing the unmarked *n-syntax* with the marked *n+syntax*), we can compare our results with previous work by Schulte im Walde (2005). Our work confirms her insight that the differences between the two versions (with vs. without reference to the syntactic functions) are visible but minimal.

6 Conclusions

This work determined the degree of compositionality of German particle verbs by two soft cluster-

ing approaches. We assumed that the more compositional a particle verb is, the more often it appears in the same cluster with its base verb, after applying a probability threshold to establish cluster membership. The overall best cluster analysis was reached by the simpler cluster approach, LSC. It could predict the degree of compositionality for 59% of the particle verbs; the correlation with the gold standard judgements was .433. Considering that we have worked with very simple distributional features that ignored other than some basic information at the syntax-semantics interface, we regard this a reasonable result. We expect that if we extended our work by incorporating the syntax-semantics transfer between particle and base verbs, we could improve on the compositionality judgements.

References

- Nadine Aldinger. 2004. Towards a Dynamic Lexicon: Predicting the Syntactic Argument Structure of Complex Verbs. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal.
- Marco Baroni and Adam Kilgarriff. 2006. Large Linguistically-processed Web Corpora for Multiple Languages. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.
- Leonard E. Baum. 1972. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities*, III:1–8.
- Nicole Dehé, Ray Jackendoff, Andrew McIntyre, and Silke Urban, editors. 2002. *Verb-Particle Explorations*. Number 1 in Interface Explorations. Mouton de Gruyter, Berlin.
- Bonnie J. Dorr and Doug Jones. 1996. Role of Word Sense Disambiguation in Lexical Acquisition: Predicting Semantics from Syntactic Cues. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 322–327, Copenhagen, Denmark.
- Afsaneh Fazly and Suzanne Stevenson. 2008. A Distributional Account of the Semantics of Multiword Expressions. *Italian Journal of Linguistics. Alessandro Lenci (guest editor): "From Context to Meaning: Distributional Models of the Lexicon in Linguistics and Cognitive Science"*, 20(1).
- Christiane Fellbaum, editor. 1998. *WordNet – An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press.
- Silvana Hartmann, Sabine Schulte im Walde, and Hans Kamp. 2008. Predicting the Degree of Compositionality of German Particle Verbs based on Empirical Syntactic and Semantic Subcategorisation Transfer Patterns. Talk at the Konvens Workshop 'Lexical-Semantic and Ontological Resources'.
- Silvana Hartmann. 2008. Einfluss syntaktischer und semantischer Subkategorisierung auf die Kompositionality von Partikelverben. Studienarbeit. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.
- Eric Joanic, Suzanne Stevenson, and David James. 2008. A General Feature Space for Automatic Verb Classification. *Natural Language Engineering*, 14(3):337–367.
- Stanislav Kavka. 2009. Compounding and Idiomatology. In Lieber and Stekauer (2009b), chapter 2, pages 19–33.
- Anna Korhonen, Yuval Krymolowski, and Zvika Marx. 2003. Clustering Polysemic Subcategorization Frame Distributions Semantically. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 64–71, Sapporo, Japan.
- Natalie Kühner. 2010. Automatische Bestimmung der Kompositionality von deutschen Partikelverben auf der Basis von Cluster-Modellen: Vergleich von LSC und PAC. Studienarbeit. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.
- Claudia Kunze. 2000. Extension and Use of GermaNet, a Lexical-Semantic Database. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 999–1002, Athens, Greece.
- Rochelle Lieber and Pavol Stekauer. 2009a. Introduction: Status and Definition of Compounding. In *The Oxford Handbook on Compounding* (Lieber and Stekauer, 2009b).
- Rochelle Lieber and Pavol Stekauer, editors. 2009b. *The Oxford Handbook of Compounding*. Oxford University Press.
- Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 17th International Conference on Computational Linguistics*, Montreal, Canada.
- Anke Lüdeling. 2001. *On German Particle Verbs and Similar Constructions in German*. Dissertations in Linguistics. CSLI Publications.
- Diana McCarthy, Bill Keller, and John Carroll. 2003. Detecting a Continuum of Compositionality in Phrasal Verbs. In *Proceedings of the ACL-SIGLEX Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan.
- Diana McCarthy, Sriram Venkatapathy, and Aravind K. Joshi. 2007. Detecting Compositionality of Verb-Object Combinations using Selectional Preferences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 369–379.
- Ines Rehbein and Josef van Genabith. 2006. German Particle Verbs and Pleonastic Prepositions. In *Proceedings of the 3rd ACL-SIGSEM Workshop on Prepositions*, pages 57–64, Trento, Italy.
- Jorma Rissanen. 1978. Modeling by Shortest Data Description. *Automatica*, 14:465–471.
- Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a Semantically Annotated Lexicon via EM-Based Clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Maryland, MD.
- Mats Rooth. 1998. Two-Dimensional Clusters in Grammatical Relations. In *Inducing Lexicons*

- with the EM Algorithm*, AIMS Report 4(3). Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.
- Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico.
- Michael Schiehlen. 2003. A Cascaded Finite-State Parser for German. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 163–166, Budapest, Hungary.
- Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging using Decision Trees. In *Proceedings of the 1st International Conference on New Methods in Language Processing*.
- Sabine Schulte im Walde, Christian Hyng, Christian Scheible, and Helmut Schmid. 2008. Combining EM Training and the MDL Principle for an Automatic Verb Classification incorporating Selectional Preferences. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 496–504, Columbus, OH.
- Sabine Schulte im Walde. 2000. Clustering Verbs Semantically According to their Alternation Behaviour. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 747–753, Saarbrücken, Germany.
- Sabine Schulte im Walde. 2004. Identification, Quantitative Description, and Preliminary Distributional Analysis of German Particle Verbs. In *Proceedings of the COLING Workshop on Enhancing and Using Electronic Dictionaries*, pages 85–88, Geneva, Switzerland.
- Sabine Schulte im Walde. 2005. Exploring Features to Identify Semantic Nearest Neighbours: A Case Study on German Particle Verbs. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 608–614, Borovets, Bulgaria.
- Sabine Schulte im Walde. 2006. Experiments on the Automatic Induction of German Semantic Verb Classes. *Computational Linguistics*, 32(2):159–194.
- Sidney Siegel and N. John Castellan. 1988. *Non-parametric Statistics for the Behavioral Sciences*. McGraw-Hill, Boston, MA.
- Barbara Stiebels. 1996. *Lexikalische Argumente und Adjunkte. Zum semantischen Beitrag von verbalen Präfixen und Partikeln*. Akademie Verlag.
- Aline Villavicencio, Francis Bond, Anna Korhonen, and Diana McCarthy. 2005. Introduction to the Special Issue on Multiword Expressions: Having a Crack at a Hard Nut. *Computer Speech and Language*, 19:365–377.

Noun Phrase Chunking and Categorization for Authoring Aids

Cerstin Mahlow and **Michael Piotrowski**

Institute of Computational Linguistics

University of Zurich

Switzerland

{mahlow, mxp}@cl.uzh.ch

Abstract

Effective authoring aids, whether for novice, second-language, or experienced writers, require linguistic knowledge. With respect to depth of analysis, authoring aids that aim to support revising and editing go beyond POS-tagging but cannot work on complete, mostly well-formed sentences to perform deep syntactic analysis, since a text undergoing revision is in a constant state of flux. In order to cope with incomplete and changing text, authoring aids for revising and editing thus have to use shallow analyses, which are fast and robust. In this paper, we discuss noun phrase chunking for German as resource for language-aware editing functions as developed in the LingURed project. We will identify requirements for resources with respect to availability, interactivity, performance and quality of results. From our experiments we also provide some information concerning ambiguity of German noun phrases.

1 Introduction

In the LingURed project¹, we are implementing functions to support writers when revising and editing German texts. For example, when a writer chooses to use a different verb, the case of the noun phrase governed by the verb may also have to be changed; since the constituents of a German noun phrase agree in case, number, and gender, the writer must move through the noun phrase and make the necessary adjustments for each word form. It frequently happens that writers forget to make some or all of the required modifications since they are focusing on the change in the verb – which may also require other modifications in distant parts of a sentence, such as the addition or deletion of a separable prefix.

Functions operating on appropriate elements reduce cognitive load and prevent errors, or *slips* (Norman, 1981), which is, in our view, preferable to hoping that a grammar checker will catch all editing and revision errors afterwards (see (Mahlow and Piotrowski, 2008; Piotrowski and Mahlow, 2009)). Note that we are not trying to make changes fully automatically, but we rather want to provide authors with “power tools” that help them make the intended edits and revisions easier and less error-prone. Authors should be in control of the text with functions helping to carry out their intentions without forcing the author to concentrate on finding the right (complex and long) sequence of low-level character-based functions.

Authoring natural language texts thus benefits from functions that operate on linguistic elements and structures and are aware of the rules of the language. We call these functions *language-aware*. Our target group are experienced writers (with respect to their knowledge of German, their writing, and their use of editors). Language-aware functions obviously require linguistic knowledge and NLP resources on different levels, as outlined by Mahlow et al. (2008).

NLP resources for use in an interactive editing environment have to meet several requirements: As we intend to support the writing *process*, the resource has to be used *interactively* – we are not interested in batch-mode systems that might be useful for some post-processing. Therefore the resource has to start and execute quickly – users will not accept to wait more than a few seconds (see (Good, 1981; Cooper et al., 2007)). As test bed we use XEmacs, which is freely available; we intend to distribute all functions freely to the community, so all resources should be *freely available*, too. The results of the resources have to be suitable for further processing. The *quality* of the results has to be high to actually support the authoring and not

¹LingURed stands for “Linguistically Supported Revising and Editing,” see <http://lingured.info>.

posing new challenges to the author or introducing errors.

Another factor influencing the design and implementation of language-aware editing functions are characteristics of the respective language – here: German. A desirable function (like pluralizing NPs) may rely on automatic unambiguous extraction of linguistic elements and determination of their morphosyntactic properties. If those elements are entirely ambiguous, it may not be possible to solve those ambiguities automatically at all – or only by using deep syntactic and semantic parsing, which is not possible during writing. Therefore it might be necessary to put the author in the loop to solve the ambiguity, which might be an easy task for humans. However, in such situation it might not be appropriate to implement such a function at all since we force the author to carry out a new task with completely different cognitive demands, thus increasing the cognitive load – i.e., the function would not fulfill what it was intended for: reducing cognitive load.

In the rest of this paper we will concentrate on a specific task – NP chunking and categorization – to be used as base for a variety of editing functions. In section 2 we will outline the requirements for a chunker and give reasons for the development of our own solution. We then show the details of our implementation and report on some experiments in section 3. Here we will also give some insights on ambiguity of German NPs as relevant basis to decide if implementation of the intended functions is possible at all. We will comment on the quality of existing annotated corpora for German and recommend to put some effort in updating them.

2 Noun phrase chunking for unrestricted German text

2.1 Motivation

Mahlow and Piotrowski (2009) outline requirements for morphological resources used for specific types of editing functions. In this paper we will concentrate on chunking for use in (a) *information functions*, (b) *movement functions*, and (c) *operations*.

A function for highlighting specific types of phrases is an example of an information function; an author may, for instance, call such a function to identify potential stylistic or grammatical trouble spots. A function for jumping to the next NP is an example of a movement function; it requires detect-

ing the next NP after the current cursor position and moving the cursor to the start of the first word of this NP. A function for modifying the grammatical case of an NP is an example of an operation: The author places the cursor on a noun and calls the operation, indicating the desired case; the operation then calls a resource to extract the needed information and makes the necessary changes to the text.

Thus, we are interested in extracting chunks to serve as resource for higher-level functions, we are not interested in checking and correcting agreement or spelling of the elements of a chunk. For information and movement functions, we have to identify the words belonging to a certain type of phrase – the usual task for a chunker. For this paper, we take into account NPs as important structural elements of natural-language text and thus a target for information functions, movement functions, or operations. For operations like pluralizing an NP, changing the case of an NP, or replacing an NP by a pronoun, we have to extract the NP and to determine the category of the phrase, i.e., the morphosyntactic properties². For German NPs these are *case*, *number*, *gender*, and *definiteness*³.

2.2 Requirements

The requirements for our NP chunking resource can be defined on the basis of general requirements for NLP resources for language-aware editing functions and on the basis of the requirements for specific purposes:

Availability For LingURed, we use the XEmacs editor⁴, which is open-source. We aim to distribute all functions we implement as open-source, too. Therefore all involved resources should be freely available.⁵

Performance The resources will be used in interactive functions and thus have to start and execute quickly.

Programming Interfaces and Further Processing The results of the chunking will be used in higher-level functions. Therefore they have

²We refer to “morphosyntactic properties” as “category,” while the process of determining this category is called “categorization.”

³In this paper, we will not further discuss definiteness, since it is relatively easy to determine.

⁴<http://xemacs.org>

⁵Unfortunately, as Mahlow and Piotrowski (2009) show, we have to make some concessions if we want to use high-quality resources.

to be delivered in a format suitable for further processing. The chunker will take input from and deliver results to a calling Emacs Lisp function, so it should offer programming interfaces to allow seamless integration.

Quality of Results The chunker should determine all NPs and deliver the correct category (case, number, and gender). The meaning of “all noun phrases” obviously depends on the definition of *noun phrase*, which we will outline in the next section.

2.3 Pragmatic definition of *noun phrases*

As for many linguistic terms, there are various definitions for the term *noun phrase*. For our purposes, we consider as *noun phrase* as sequence of word forms consisting of a noun preceded by one or more adjectives and/or a determiner. Usually, this type of NPs is called *base NP*, *non-recursive NP*, *noun kernel*, or *contiguous NP* and follows the definition of the CoNLL-2000 chunking shared task (Tjong Kim Sang and Buchholz, 2000). We do not consider NPs consisting only of a single noun here, since determining the category of a noun only involves the morphological analyzer.

For example, in the sentence *Der Traum vom Essen ohne Reue beschert der Nahrungsmittelindustrie schöne Perspektiven*. (‘The dream of eating without regrets gives great prospects to the food industry.’)⁶, we would like to extract the NPs as marked in example 1. In particular, we do not aim to extract recursive NPs.

- (1) [NP Der Traum] [NP vom Essen] ohne
 [N Reue] beschert [NP der
 Nahrungsmittelindustrie] [NP schöne
 Perspektiven].

Note that we mark *vom Essen* as NP although it contains a preposition. Since *vom* is a merged word form consisting of a preposition (*von*) and a determiner (*dem*), we will be able to split this word form, strip the preposition, and thus get the NP *dem Essen*.

We concentrate on extracting contiguous base NPs for two reasons. First, there is a simple test to determine what to include in an NP when considering changing case or number of an NP: All word

⁶Unless stated differently, all examples are taken from a corpus of the German newspaper “Der Tagesspiegel. Zeitung für Berlin und Deutschland” from 2005 and 2006, consisting of 2,235,726 word forms (133,056 sentences).

forms not affected by the change do not belong to the NP. In German, it is possible to embed complex phrases into an NP, as in *eine für die Verhältnisse hohe Qualität* (‘a high quality with respect to the circumstances’, literally: ‘a for the circumstances high quality’):

- (2) [NP eine [PP für [NP die Verhältnisse]] hohe
 Qualität]

Applying our simple test, it would be necessary to extract the discontinuous base NP *eine hohe Qualität*. Kübler et al. (2010) introduce the *stranded noun chunk* (sNX) for the determiner *eine* to be able to mark the desired NP. However, it involves deep syntactic analysis to automatically annotate such phrases correctly. And this involves the second reason to concentrate on contiguous NPs: In the LingURed project, we are dealing with *texts in progress*; the text is not finished and therefore some parts of the texts will always be ill-formed, incomplete, or inconsistent. These “three I’s,” as Van De Vanter (1995, p. 255) calls them, hinder deep syntactic analysis and make it very hard to determine discontiguous NPs reliably.

Sequences of adjectives may be interrupted by conjunctions (the STTS tag KON) or adverbs (ADV) (including adjectives used adverbially). The role of the determiner can be filled by definite determiners (ART), indefinite determiners (ART), prepositions with determiner (APPRART), possessive pronouns (PPOSAT), attributive indefinite pronouns with and without determiner (PIDAT and PIAT), and attributive demonstrative pronouns (PDAT). We do not consider proper names as nouns. The following list shows some examples:

- (3) [ART Eine] gemischte Crew
 ‘a mixed crew’
 [ART der] transatlantischen Fusion
 ‘of the transatlantic fusion’
 [APPRART beim] Sozialminister
 ‘at the minister of social affairs’
 [PPOSAT unserem] zeitgeschichtlichen Be-
 wusstsein
 ‘our sense of contemporary history’ (dative)
 [PIDAT beide] Polizisten
 ‘both policemen’
 [ART die] [PIDAT beiden] Polizisten
 ‘these two policemen’
 [PIAT einige] Automobilhersteller
 ‘some car manufacturers’

[_{PDAT} diese] heiklen Verfahren
 ‘these critical processes’

[_{PPSAT} seines] [_{ADV} besonders] religiösen
 [_{KON} oder] [_{ADV} besonders] homosexuellen
 Gehalts
 ‘of its especially religious or especially ho-
 mosexual content’

2.4 Related work

A number of chunkers for German are described in the literature (e.g., (Schmid and Schulte im Walde, 2000; Kermes and Evert, 2002; Schiehlen, 2002); see Hinrichs (2005) for an overview). However, all systems we know of are primarily intended for batch processing, not interactive use. For example, the TreeTagger chunker (Schmid, 1995) is frequently used for German, but it is not designed to be used interactively and is thus not suitable for our purposes.

Furthermore, since chunking is typically used in applications such as information extraction or information retrieval, the focus is on the identification of NPs, not on their categorization. Although many noun chunkers make use of morphological information to determine the extent of chunks (see (Church, 1988; Ramshaw and Marcus, 1995; Schiehlen, 2002)), they usually do not deliver the category of the NPs.

The exact definition of an NP also varies and clearly depends on the intended application; for example, the TreeTagger chunker uses a definition similar⁷ to ours (Schmid and Schulte im Walde, 2000); YAC (Kermes and Evert, 2002), on the other hand, is intended for corpus preprocessing and querying and outputs recursive chunks.

After considering the common algorithms and approaches and our specific requirements, we decided to implement our own NP chunker using low-level resources already used for other functions in the LingURed project. We will describe our implementation and evaluation experiments in the next section.

3 The NPcat Chunker

For the LingURed project, we decided to implement an NP chunker to identify NPs and determine their categories according to the definition of NPs given above. The implementation is called *NPcat*

⁷However, besides noun chunks, it also outputs prepositional chunks (PCs). A PC consists of a preposition and an NP. Since the NP is not marked explicitly, some post-processing would be required to also extract these NPs.

and is based on part-of-speech tagging and morphological analysis.

For tagging we use the Mbt part-of-speech tagger (Daelemans et al., 2010). Piotrowski and Mahlow (2009) have shown that it can be integrated easily into XEmacs. The quality of the tagging results obviously depends on the quality of the training corpus Mbt is trained on. We will discuss this issue in section 3.2.1. For the work described in this paper, we have trained Mbt on TüBa-D/Z (Tübinger Baumbank des Deutschen/Schriftsprache), release 5 (Telljohann et al., 2009).

As a morphological resource we use GERTWOL (Koskenniemi and Haapalaisten, 1996). As Mahlow and Piotrowski (2009) show, it is currently the only morphological system for German available⁸ that meets the requirements for integration into real-world applications and delivers high-quality results. GERTWOL is shipped as shared library with a C API for integration into applications.

Both Mbt and GERTWOL are already successfully used for other language-aware editing functions in the LingURed project.

3.1 Implementation details

NPcat uses three steps, executed successively, to obtain the NPs and their categories:

1. Determine the POS of all word forms in a (span of) text using Mbt.
2. Extract NPs matching our definition given in section 2.3.
3. Categorize all elements of an NP using GERTWOL and determine the possible categories of the NP (since the elements must agree in case, number, and gender, this can be described as the intersection of the categories of the constituents).

As an example, let us consider the following sentence: *Nur wenn dieses strikte Verbot gelockert werde, heiSt es in einer Studie der DG-Bank, könne über eine bessere Aufklärung der Verbraucher das brachliegende Potenzial konsequent erschlossen werden.* (‘Only if this strict ban were lifted, a study of DG-Bank says, the untapped potential could systematically be exploited through better counseling of consumers’). Mbt delivers the tags

⁸It is not open source, but an academic license is available for a reasonable fee.

presented in (4) below. Note that *gelockert*, *DG-Bank* and *Potenzial* are not in the lexicon, and the unknown words case base was used to predict the tags. We use the tags from the Stuttgart-Tübingen Tagset (STTS) (Schiller et al., 1999).

- (4) [ADV Nur] [KOUS wenn] [PDAT dieses]
 [ADJA strikte] [NN Verbot] [VVPP gelockert]
 [VAFIN werde] [\$. ,] [VVFIN heiSSt] [PPER es]
 [APPR in] [ART einer] [NN Studie] [ART der]
 [NN DG-Bank] [\$. ,] [VMFIN könne] [APPR über]
 [ART eine] [ADJA bessere] [NN Aufklärung]
 [ART der] [NN Verbraucher] [ART das]
 [ADJA brachliegende] [NN Potenzial]
 [ADJD konsequent] [VVPP erschlossen]
 [VAINF werden] [\$. .]

The following NPs are then extracted from this sentence:

- (5) a. dieses strikte Verbot
 b. einer Studie
 c. der DG-Bank
 d. eine bessere Aufklärung
 e. der Verbraucher
 f. das brachliegende Potenzial

In the third step, the word forms in each NP are analyzed morphologically by GERTWOL. For (5a), GERTWOL delivers the analyses shown in listing 1. We ignore the analyses for parts-of-speech that cannot be part of an NP – in this case, the pronoun readings for *dieser* and the verb readings for *Verbot*.

With this information, NPcat tries to determine the category of the NP. The elements of an NP have to agree with respect to case, number, and gender. The gender for *Verbot* is neuter, thus the readings as feminine and masculine for the adjective and the masculine reading for the determiner are excluded. The readings for the determiner and the noun are singular only, thus we can exclude the plural readings for the adjective. The values for gender and number are thus: Neuter and singular. There are only two corresponding readings for the adjective (nominative and accusative singular neuter), both readings are possible for the determiner and the noun as well – so we get two possible categories for the phrase *dieses strikte Verbot*: Nominative singular neuter and accusative singular neuter.

From this example we can conclude: (a) As the elements of a German NP agree with respect to

```

dieses
(
("dieser" . [PRON MASC SG GEN])
("dieser" . [PRON NEU SG NOM])
("dieser" . [PRON NEU SG ACC])
("dieser" . [PRON NEU SG GEN])
("dieser" . [DET MASC SG GEN])
("dieser" . [DET NEU SG NOM])
("dieser" . [DET NEU SG ACC])
("dieser" . [DET NEU SG GEN])
)
strikte
(
("strikt" . [ADJ FEM SG NOM POS])
("strikt" . [ADJ FEM SG ACC POS])
("strikt" . [ADJ PL NOM POS])
("strikt" . [ADJ PL ACC POS])
("strikt" . [ADJ MASC SG NOM POS])
("strikt" . [ADJ NEU SG NOM POS])
("strikt" . [ADJ NEU SG ACC POS])
("strikt" . [ADJ FEM SG NOM POS])
("strikt" . [ADJ FEM SG ACC POS])
)
Verbot
(
("Ver|bot" . [N NEU SG NOM])
("Ver|bot" . [N NEU SG ACC])
("Ver|bot" . [N NEU SG DAT])
("ver|biet-en" . [V PAST IND SG1])
("ver|biet-en" . [V PAST IND SG3])
)
```

Listing 1: Analyses for the word forms in *dieses strikte Verbot* by GERTWOL

case, number, and gender, we can use the intersection of the categories of those word forms to determine the category of the NP. (b) German NPs can be ambiguous concerning their morphosyntactical properties. We will have a closer look at this phenomenon in section 3.2.3.

3.2 Experiments

To evaluate the appropriateness of our approach, we carried out some experiments. Some of these experiments were also intended to get an impression of morphosyntactic features of German NPs, in order to decide whether functions involving extracting NPs and determining their category can be of any use at all. The quality of the results delivered by NPcat clearly depends on the quality of the tagging and the quality of the morphological analysis.

3.2.1 Quality of the tagging

We decided to use Mbt for tagging as it is open-source software and can be used interactively. When using Mbt, it has to be trained on an annotated corpus. The currently available annotated corpora for German with an appropriate size to be used as training set are NEGRA, TIGER, and TüBa-D/Z. Of these, TüBa-D/Z is being actively maintained and enhanced. However, all of these corpora contain almost exclusively texts written according

to spelling rules *before* the 1996 spelling reform. There seem to be some articles in the TIGER written according to current spelling rules. However, this is not mentioned in the release notes. Both NEGRA and TüBa-D/Z do not include texts written according to current spelling rules. Thus, these corpora do not represent the *current* spelling and are, strictly speaking, not suitable to be used as resource for any application dealing with current texts.

To our knowledge there is only one annotated resource available written in current German spelling: The two small German corpora in the SMULTRON treebank (Gustafson-Čapková et al., 2007). However, with around 520 sentences each⁹, they are too small to serve as a resource for training Mbt. They also lack morphological information (there is information on gender only) and thus cannot be used as a gold standard for morphological analysis and NP categories.

In the TIGER corpus, no difference is made between attributive indefinite pronouns with and without determiner. However, this distinction is essential for our definition of NPs: Word forms tagged as PIAT (attributive indefinite pronoun without determiner) like *kein* ('none') cannot be preceded by a determiner, whereas word forms tagged as PIDAT (attributive indefinite pronoun with determiner) can be preceded by a determiner, e.g., *die beiden Polizisten* ('the two policemen'). PIAT-tagged word forms, as well as PIDAT-tagged word forms can fill the determiner slot. However, if there is a determiner preceding a PIDAT-tagged word form, it has to be included into the NP, and the PIDAT-tagged word form will then be inflected like an adjective. Using TIGER will thus introduce errors in determining NPs.

We eventually decided to use TüBa-D/Z for training Mbt, since it is the largest corpus, it is actively maintained, and differentiates between PIAT and PIDAT.

3.2.2 Quality of noun chunks

Given a tagged text, how many of the NPs (as defined in section 2.3) are actually found by NPcat, and how many of them are correct?

As noted above, this primarily depends on the quality of the POS tagging – clearly, if a noun is mistagged as a verb, our rules cannot find the cor-

⁹7,416 tokens (529 sentences) taken from the novel "Sophie's World" and 10,987 tokens (518 sentences) taken from three business texts.

responding NP. The question is thus how well the tagger is able to identify the constituents of NPs; this question is not answered by general accuracy numbers, but would require comparison to a gold standard. While annotated corpora usually include annotations for NPs or noun chunks, the underlying definition of noun chunks does not necessarily correspond to our definition. We would thus have to create a gold standard ourselves – something we have not yet done at the time of this writing, thus we cannot provide evaluation results for this aspect.

3.2.3 Categories of noun chunks

For our application, the categorization of NPs is the most critical aspect, since writers should neither be irritated by incorrect analyses nor bothered by unnecessary queries from the system.

Evert (2004) showed that only about 7% of German nouns can be categorized unambiguously in isolation. He found that around 20% of German nouns can be categorized unambiguously when taking into account some syntactical processing – when using the left context of a noun, i.e., adjectives and determiners.

We ran NPcat on a corpus of articles from the German newspaper "Der Tagesspiegel" from 2005 and 2006, consisting of 2,235,726 word forms (133,056 sentences). NPcat found 516,372 NPs, 152,801 of them consisted of a single noun only and were thus excluded after step 2. When looking at unique NPs, we found 245,907 NPs, of which 45,029 were single nouns. Table 1 shows the categorization results for all NPs and for unique NPs (excluding single nouns).

NPcat marks NPs as "unknown" in the following cases:

- No agreement between the elements of a potential NP (e.g., *alle Auto* 'all car')
- Tags delivered by Mbt are wrong (e.g., *kniend* 'kneeling' tagged as noun)
- A word form is misspelt and thus not recognized by GERTWOL, although tagged correctly by Mbt (e.g., *Rathuas* instead of *Rathaus* 'city hall')
- The NP is correct, but some words are not recognized by GERTWOL (e.g., *schwächelnden* 'flagging' in *der schwächelnden US-Konjunktur* 'of the flagging US economy')

	Total	Unknown	1 category	2 categories	3 categories	4 or more
All NPs	363571	16827 (4.63%)	136444 (37.53%)	181838 (50.01%)	7745 (2.13%)	20717 (5.70%)
Unique NPs	200878	14506 (7.22%)	71420 (35.55%)	94893 (47.24%)	4636 (2.31%)	15423 (7.68%)

Table 1: Categories of NPs

The results show that more than 35% of the NPs can be categorized unambiguously, and for another 50% two categories are found.¹⁰ This is a quite satisfying result with respect to our ultimate purpose of using NPcat as a resource for interactive editing functions. These functions are intended to reduce cognitive load and make editing and revising easier; ambiguous intermediate results of NLP resources may require interaction with the user, which could be counterproductive.

Our experiment shows that no interaction is needed in one third of all cases involving NPs. For NPs with two categories (about half of all NPs), the need for interaction depends on the desired operation and the morphosyntactical properties (including inflection class) of the NP and cannot be determined beforehand. To our knowledge, there is currently no research on these properties of German NPs.¹¹

For example, when pluralizing an NP, the plural forms of the constituent words of the NP have to be generated, preserving gender and case. For *das Konzerthaus* ('the concert hall') we obtain two categories: NEU SG NOM and NEU SG ACC. The plural forms of these categories share the same surface, *die Konzerthäuser* – thus, even though the category is ambiguous, no interaction with the user would be needed in this case. 29,433 of all NPs (8.1%) in our test corpus were categorized as NEU SG NOM and NEU SG ACC.

For *der Reparaturwerkstatt* ('to/of the garage') we obtain the two categories FEM SG GEN and FEM SG DAT. The plural forms of these categories are *der Reparaturwerkstätten* and *den Reparaturwerkstätten* – here, the user either has to identify the category of the original NP or has to choose be-

¹⁰It might be possible to reduce the number of ambiguous NPs considering verb frames. However, this would involve deeper syntactic analysis, for subordinate clauses the verb might even not yet be written when the author calls an NP-based function.

¹¹There is an open field for further research questions like the ratio between contiguous and discontiguous NPs or the ratio between simple and complex NPs, as one of the reviewers proposed. Kübler et al. (2010) report some first insights concerning embedded adjective phrases in NPs within TüBa-D/Z. More work in this area is clearly needed, but it is not in the focus of this paper or the LingURed project as such.

tween the two possible plural NPs. 41,802 of all NPs (11.5%) are categorized as FEM SG GEN and FEM SG DAT.

On the basis of the experimental results and these considerations, we believe it is reasonable to assume that no interaction is needed in more than 60% of all cases.

3.2.4 Quality of categorization

Finally, which quality can we expect for the categories of the identified NPs? The ambiguity of NPs clearly influences the interaction with the user when used in operations as shown above. However, we need some confidence about the correctness of the determined category of a certain NP, since users should know whether they can trust the changes made by operations based on NP chunking. If the correctness is insufficient, users would have to check – and possibly revise – all changes and there would be no benefit in using such an operation.

To answer this question, we randomly chose two samples – one from the unambiguous and one from the two-fold ambiguous NPs of the unique NPs –, each consisting of 384 NPs. The sample size n was chosen to achieve a confidence level of 95% with a 5% error, according to the standard formula

$$n = \frac{Z^2 \sigma^2}{e^2}$$

where $Z^2 = 1.96$ for a confidence level of 95%, e is the desired level of precision (we use a confidence interval of 5%), and σ^2 is the variance of the population (we assume $\sigma^2 = .25$ for maximum variability).

The samples were then manually checked. We found that the categories for non-ambiguous NPs were almost all correct; there were only two false categories:

- (6) a. * *deren Freundin*: FEM SG GEN
‘whose girlfriend’
- b. * *deren Schwester*: FEM SG GEN
‘whose sister’

In both cases, *deren* ('whose') was incorrectly tagged as PDAT instead of as PRELAT. In fact, both

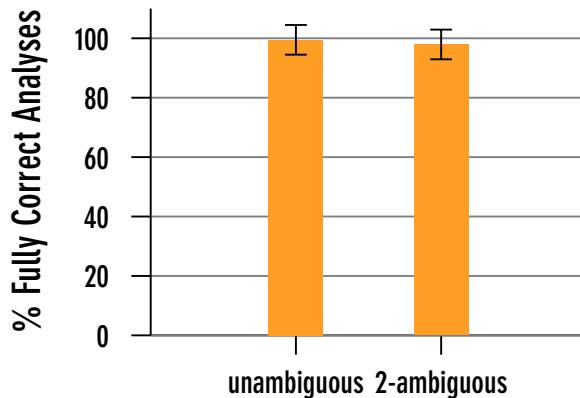


Figure 1: Percentage of completely correct analyses with a confidence interval of 5%

NPs are ambiguous with respect to case. This type of problem may be reduced by improving the training of the tagger.

Incorrect categories for two-fold ambiguous NPs are due to unusual analyses of the respective noun by GERTWOL as listed in (7). If GERTWOL used some kind of weighting, unlikely decompositions like *Flugzeuge < der Flug-Zeuge* (7a) or *Urteil < der Ur-Teil* (7b), or readings as nominalized verbs like *Hauptsätzen < das Haupt-Sätzen* (7e) could be avoided.

- (7) a. * der Zivilflugzeuge: MASC SG NOM, NEU PL GEN
‘(of) the airplanes’
- b. * seinem Urteil: MASC SG DAT, NEU SG DAT
‘his decision’
- c. * vielen StraSSenkämpfen: MASC PL DAT, NEU SG DAT
‘many riots’
- d. * möglichen Punkten: MASC PL DAT, NEU SG DAT
‘possible points’
- e. * kurzen Hauptsätzen: MASC PL DAT, NEU SG DAT
‘short main clauses’

4 Conclusion

Interactive editing applications pose specific challenges to NLP resources, which sometimes differ significantly from those posed by non-interactive applications.

In this paper, we outlined requirements for an NP chunker and categorizer to be used as resource for

language-aware editing functions to support authoring of German texts. Currently available chunkers do not meet these requirements and we therefore had to implement our own solution – NPcat – on the basis of existing resources for tagging and morphological analysis. We showed that NPcat meets the usual quality criteria for NP chunking of German texts.

On the one hand, our experiments showed that NPcat is able to categorize NPs with a high degree of correctness. On the other hand, we found that there is an urgent need to put effort in updating existing annotated corpora for German – or creating new ones – to allow processing of current texts written according to current spelling rules: It is evident that the performance of a tagger trained on text in the pre-1996 orthography is suboptimal when applied to text written in the post-1996 orthography.

When we started the LingURed project, we argued that in the first decade of the 21st century it is finally possible to successfully develop editing functions based on NLP resources. First attempts in the 1980s and 1990s were not successful, since the NLP resources available at that time were still immature and the limited computing power made interactive NLP applications almost impossible. Since then, computers have become much faster and provide for very fast execution of NLP tools. However, while performance is no longer a problem, NLP systems for German still do not meet our expectations with respect to maturity and quality of results. Mahlow and Piotrowski (2009) have shown that the situation with respect to morphological analysis and generation for German is disappointing: There is, in effect, only one system available (GERTWOL), and it is not open source. With respect to chunking, we find that the situation is very similar.

References

- Kenneth W. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143, Morristown, NJ, USA. Association for Computational Linguistics.
- Alan Cooper, Robert Reimann, and David Cronin. 2007. *About Face 3: The Essentials of Interaction Design*. Wiley, Indianapolis, IN, USA, 3rd edition.

- Walter Daelemans, Jakub Zavrel, Antal van den Bosch, and Ko van der Sloot. 2010. MBT: Memory-Based Tagger version 3.2 reference guide. Technical report, Induction of Linguistic Knowledge Research Group, Department of Communication and Information Sciences, Tilburg University, June.
- Stefan Evert. 2004. The statistical analysis of morphosyntactic distributions. In *LREC 2004 Fourth International Conference on Language Resources and Evaluation*, pages 1539–1542.
- Michael Good. 1981. Etude and the folklore of user interface design. In *Proceedings of the ACM SIGPLAN SIGOA symposium on Text manipulation*, pages 34–43, New York, NY, USA. ACM.
- Sofia Gustafson-Čapková, Yvonne Samuelsson, and Martin Volk. 2007. SMULTRON (version 1.0) – The Stockholm MULTilingual parallel TReebank.
- Erhard W. Hinrichs. 2005. Finite-state parsing of German. In Antti Arppe, Lauri Carlson, Krister Lindén, Jussi Piitulainen, Mickael Suominen, Martti Vainio, Hanna Westerlund, and Anssi Yli-Jyrä, editors, *Inquiries into Words, Constraints, and Contexts. Festschrift for Kimmo Koskenniemi on his 60th Birthday*, CSLI Studies in Computational Linguistics ONLINE, pages 35–44. CSLI Publications, Stanford, CA, USA.
- Hannah Kermes and Stefan Evert. 2002. YAC – a recursive chunker for unrestricted German text. In M. G. Rodriguez and C. P. Araujo, editors, *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 1805–1812.
- Kimmo Koskenniemi and Mariikka Haapalaisten. 1996. GERTWOL – Lingsoft Oy. In Roland Hausser, editor, *Linguistische Verifikation. Dokumentation zur Ersten Morpholympics 1994*, chapter 11, pages 121–140. Niemeyer, Tübingen.
- Sandra Kübler, Kathrin Beck, Erhard Hinrichs, and Heike Telljohann. 2010. Chunking German: An unsolved problem. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 147–151, Uppsala, Sweden, July. Association for Computational Linguistics.
- Cerstin Mahlow and Michael Piotrowski. 2008. Linguistic support for revising and editing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 9th International Conference, CICLing 2008, Haifa, Israel, February 17–23, 2008. Proceedings*, pages 631–642, Heidelberg. Springer.
- Cerstin Mahlow and Michael Piotrowski. 2009. A target-driven evaluation of morphological components for German. In Simon Clematide, Manfred Klenner, and Martin Volk, editors, *Searching Answers – Festschrift in Honour of Michael Hess on the Occasion of his 60th Birthday*, pages 85–99. MV-Verlag, Münster, October.
- Cerstin Mahlow, Michael Piotrowski, and Michael Hess. 2008. Language-aware text editing. In Robert Dale, Aurélien Max, and Michael Zock, editors, *LREC 2008 Workshop on NLP Resources, Algorithms and Tools for Authoring Aids*, pages 9–13, Marrakech, Morocco. ELRA.
- Donald A. Norman. 1981. Categorization of action slips. *Psychological Review*, 88:1–15.
- Michael Piotrowski and Cerstin Mahlow. 2009. Linguistic editing support. In *DocEng’09: Proceedings of the 2009 ACM Symposium on Document Engineering*, pages 214–217, New York, NY, USA, September. ACM.
- Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey. Association for Computational Linguistics.
- Michael Schiehlen. 2002. Experiments in German noun chunking. In *Proceedings of the 19th international conference on Computational Linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.
- Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Stuttgart.
- Helmut Schmid and Sabine Schulte im Walde. 2000. Robust German noun chunking with a probabilistic context-free grammar. In *Proceedings of the 18th conference on Computational linguistics*, pages 726–732, Morristown, NJ, USA. Association for Computational Linguistics.
- Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.
- Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2009. Stylebook for the tübingen treebank of written German (TüBa-D/Z). Technical report, Universität Tübingen, Seminar für Sprachwissenschaft.
- Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*, pages 127–132, Morristown, NJ, USA. Association for Computational Linguistics.
- Michael Lee Van De Vanter. 1995. Practical language-based editing for software engineers. In *Software Engineering and Human-Computer Interaction, Lecture Notes in Computer Science*, pages 251–267. Springer.

Learning meanings of words and constructions, grounded in a virtual game

Hilke Reckman

The Media Laboratory
MIT
USA

reckman@media.mit.edu

Jeff Orkin

The Media Laboratory
MIT
USA

jorkin@media.mit.edu

Deb Roy

The Media Laboratory
MIT
USA

dkroy@media.mit.edu

Abstract

We discuss the use of data from a virtual world game for automated learning of words and grammatical constructions and their meanings. The language data is an integral part of the social interaction in the game and consists of chat dialogue, which is only constrained by the cultural context, as set by the nature of the provided virtual environment. This paper presents a preliminary exploration of syntactic and semantic aspects of the dialogue in the corpus. We show how simple association metrics can be used to extract words, phrases and more abstract syntactic patterns with targeted meanings or speech-act functions, by making use of the non-linguistic context.

1 Introduction

The use of corpora has proven to be of great value to natural language processing tasks. Parsers for syntactic analysis, for example, have become highly robust and fairly accurate. Advanced semantic processing, however, remains a great challenge. Although applications involving complex use of natural language, such as question answering (Dang et al., 2007), have been shown to profit from deep semantic processing and automated reasoning, a major bottleneck for such techniques, now that several robustness issues have been addressed, appears to be a lack of world knowledge (Giampiccolo et al., 2007). This is not too surprising, since the corpora used are nearly always either text-only or text with some level of, usually task-specific, initially human, annotation. Therefore NLP programs generally have no access at all to non-linguistic context.

A way to get at meaning more naturally is through grounded data and/or grounded interac-

tion, as our own knowledge of natural language meanings is thought to be grounded in action and perception (Roy, 2005). Viewing language as a complex adaptive system which evolves in a community through grounded interaction can yield important new insights (e.g. (Steels, 2003)).

Whereas the techniques for real-world perception in computers are still rather limited, virtual worlds are getting ever more complex and realistic, have many visitors, and do not share the perceptual challenges. This offers great potential for data collection¹. Examples of virtual word learning-through-interaction projects involving language and/or social behavior are ‘Wubble World’ (Hewlett et al., 2007) and ‘Agent Max’ (Kopp et al., 2003).

Our research focuses on learning from data, rather than through interaction, though the latter may be possible in a later stage of the project. We aim at developing algorithms that learn the meanings of words and grammatical constructions in human language in a grounded way. Our data consists of game-logs from the ‘Restaurant Game’ (Orkin and Roy, 2007), which is an on-line 2-player game in which human players play the roles of customer and waitress in a virtual restaurant. The dataset includes both what they do, and what they say to each other (through chat). It is thus a collection of episodes that take place in a virtual restaurant, enacted by human players, and it has already been shown that useful knowledge about typical activities at restaurants can be extracted from these data. The intuition is that a human student of English starting from scratch (but with some common sense knowledge about how

¹von Ahn & Dabbish (2004) were among the first to realize the potential of collecting human knowledge data on-line, in a game setup, collecting a large image-labeling corpus.

things go in restaurants), could learn quite a bit of English from studying these episodes; possibly enough to play the game. We try to computationally simulate such a learning process. One of the overarching questions underlying this work is what knowledge about language and how it works is needed to extract knowledge about constructions and their meanings from grounded data.

Although the things people say and the things people do tend to be closely related in the restaurant game scenes, the relation is not as straightforward as in some related work, where the data was much more restricted, and the language part contained only descriptions (Gorniak and Roy, 2004) or only directives (Fleischman and Roy, 2005; Gorniak and Roy, 2005). The datasets of those experiments were designed purely for learning word meaning, with each utterance being a nearly direct description of its accompanying action. The Restaurant Game on the other hand was designed for learning natural restaurant behavior, including language, to animate artificially-intelligent characters who can play the game in a convincing, human-like way, and therefore the interaction is much more open-ended. This makes the learning of language from our data a different type of challenge.

In this paper we first introduce The Restaurant Game in section 2, then we explain our main method for extracting words based on their associations with objects in section 3. Next, in section 4, we zoom in on the items on the menu, and extract words and also multi-word units referring to them. This in turn allows us to extract sentence patterns used for ordering food (section 6). Finally we attempt to find words for food items that are not on the menu, by using these patterns, and wrap up with a concluding section.

2 The Restaurant Game

The restaurant theme was inspired on the idea of Schank & Abelson (1977), who argued that the understanding of language requires the representation of common ground for everyday scenarios. Orkin & Roy (2007) showed in The Restaurant Game Project that current computer game technology allows for simulating a restaurant at a high level-of-detail, and exploit the game-play experiences of thousands of players to capture a wider coverage of knowledge than what could be hand-crafted by a team of researchers. The goal is au-



Figure 1: screen-shot from the Restaurant Game, waitress's perspective

tomating characters with learned behavior and dialogue. The ongoing Restaurant Game project has provided a rich dataset for linguistic and AI research. In an on-line multi-player game humans are anonymously paired on-line to play the roles of customers and waitresses in a virtual restaurant (<http://theRestaurantGame.net>). Players can chat with open-ended typed text, move around the 3D environment, and manipulate 47 types of interactive objects through a point-and-click interface. Every object provides the same interaction options: pick up, put down, give, inspect, sit on, eat, and touch. Objects respond to these actions in different ways. For instance, food diminishes bite by bite when eaten, while eating a chair makes a crunch sound, but does not change the shape of the chair. The chef and bartender are hard-coded to produce food items based on keywords in chat text. A game takes about 10-15 minutes to play. Everything players say and do is logged in time-coded text files on our servers. Player interactions vary greatly, and while many players do misbehave, Orkin and Roy (2007) have demonstrated that enough people do engage in common behavior that it is possible for an automatic system to learn statistical models of typical behavior and language that correlate highly with human judgment of typicality.

Previous research results include a learned plan-network that combines action and language in a statistical model of common ground that associates relevant utterances with semantic context and a first implementation of a planner that drives AI characters playing the game (Orkin and Roy, 2009).

A total of 10.000 games will be collected, of which over 9000 have been collected already. The average game consists of 85 physical actions and 165 words, contained in 40 lines of dialogue.

Our analyses in this paper are based on a randomly selected set of 1000 games, containing a total of 196,681 words (8796 unique words). This is not a huge amount, but it yields fairly robust results, because we are working with a coherent domain. Of course there will always be utterances that our system cannot make sense of, because sometimes players talk about things that have nothing to do with the game.

The dialogue is grounded in two (partially overlapping) ways. Not only is there a simulated physical environment with objects that can be manipulated in various ways, but also social patterns of reoccurring events provide an anchor for making sense of the dialogue.

3 Associations between objects and words

The game contains a number of different objects, and trying to find words that are used to refer to these is a natural place to start. Let us start out with a simple assumption and see how far it gets us: We expect that objects are most talked about around the times when they are involved in actions. This means we can measure association strength in terms of relative co-occurrence. This is how collocational, or more generally, collostructional strength is commonly measured (Stefanowitsch and Gries, 2003): How often do two things co-occur compared to how often each of them occurs in total? The Chi square (χ^2) value is a good measure of that (Manning and Schütze, 2000).

Game logs were processed as follows. All actions between two lines of dialogue were treated as one action block. All lines of dialogue between two actions were treated as one dialogue block. For each action block it was noted which objects it contains, for each dialogue block which words it contains. Then for each object its association strength with each word was computed based on the occurrence of that word in the blocks of dialogue immediately preceding the blocks containing the object. Preceding dialogue turned out to work better than following dialogue, which might be due to the nature of the corpus with relatively many requests and directives. Only positive associations were taken into account, that is cases

where the observed co-occurrence was higher than the co-occurrence that would be expected if words and objects were distributed randomly over the game. In other words, we compare the portion that a word makes up in an object's preceding-block-context to the portion it makes up in the total corpus. The phi value, derived from χ^2 was used as a metric of association strength. We applied basic smoothing (absolute discounting), and required that items occur in at least 4 games in the corpus, to be scored. This reduces noise created by a particular player repeating the same atypical thing a number of times in a game. Table 1 shows all object types with their 5 most strongly correlated words in the preceding dialogue block.

We see that in spite of the simple approach, many objects correlate most strongly with sensible words (we are at this point primarily interested in referring words and phrases). Words for ordered food and drink items are picked up well, as well as those for the menu, the bill, vase and flowers. Some of the kitchen utensils such as the pot and pan are not used often and systematically enough to give good results in this first rough method. When objects are clustered on the basis of their physical interactions, these objects also fail to cluster due to sparse data (Orkin, 2007). The furniture items seem to mostly associate with words for items that are put on them. Looking into the actions in some more detail seems to be needed here, but remains for future work. Relevant context can of course extend beyond the preceding block. We will use a modified notion of context in the next sections.

Since the assumption we made about co-occurrence is so general, we expect it to apply to other domains too: frequently used movable items will most likely pair up with their referring words quite well.

We have observed that in many cases sensible words show up as (most) strongly associated with the objects, but we have no way yet to determine which are the referring ones, which are otherwise related and which are unrelated. Some objects can be referred to by different synonymous words such as ‘bill’ and ‘check’. Others can be referred to by a phrase of more than one word, such as ‘spaghetti marinara’. We need to be able to distinguish those cases. The issue is addressed in the following section.

object	word 1	phi w1	word 2	phi w2	word 3	phi w3	word 4	phi w4	word 5	phi w5
WATER	water	0.24	please	0.02	glass	0.02	thank	0.01	of	0.01
TEA	tea	0.34	te	0.05	pie	0.02	cup	0.01	t	0.01
COFFEE	coffee	0.22	coffe	0.03	cup	0.02	tu	0.01	please	0.01
BEER	beer	0.26	beers	0.03	berr	0.02	please	0.02	give	0.02
REDWINE	red	0.23	wine	0.12	redwine	0.02	wines	0.02	glass	0.01
WHITEWINE	white	0.20	wine	0.09	whine	0.02	red	0.02	degree	0.02
SOUP	soup	0.21	vegetable	0.05	jour	0.04	de	0.03	du	0.03
SALAD	salad	0.17	cobb	0.09	cake	0.02	cob	0.02	steak	0.02
SPAGHETTI	spaghetti	0.18	spagetti	0.08	marinara	0.04	pasta	0.04	steak	0.02
FILET	steak	0.25	filet	0.14	mignon	0.08	lobster	0.03	salad	0.03
SALMON	salmon	0.15	grilled	0.05	fish	0.05	steak	0.01	idiot	0.01
LOBSTER	lobster	0.19	steak	0.03	thermador	0.03	cake	0.02	salad	0.02
CHEESECAKE	cheesecake	0.15	cake	0.13	cheese	0.08	cherry	0.05	cheescake	0.05
PIE	pie	0.35	berry	0.07	cake	0.03	steak	0.02	tea	0.02
TART	tart	0.21	nectarine	0.08	tarts	0.01	coffee	0.01		
MENU	menu	0.08	seat	0.03	start	0.02	please	0.02	soup	0.02
BILL	bill	0.08	check	0.07	pay	0.04	thank	0.03	again	0.02
VASEOFFLOWERS	flowers	0.04	these	0.02	flower	0.01	vase	0.01	roof	0.01
BOWLOFFRUIT	fruit	0.05	fruits	0.03	bowl	0.02	vase	0.01	serious	0.01
BOTTLEOFWATER	bottle	0.01	water	0.01	cold	0.01	ass	0.01	!	0.01
BOTTLEOFWINE	bottle	0.03	wine	0.02	brandy	0.02	\$50	0.02	dead	0.01
BOTTLEOFBRANDY	brandy	0.02	woman	0.02	cake	0.01	whiskey	0.01	road	0.01
BINOFTRASH	trash	0.05	garbage	0.02	cops	0.02	lmao	0.01	bin	0.01
POT	hit	0.02	pot	0.02	stuck	0.02	wanna	0.01	yup	0.01
PAN	kitchen	0.01	move	0.01	fish	0.01	an	0.00	off	0.00
MICROWAVE	microwave	0.06	kitchen	0.02	break	0.01	staff	0.01	ha	0.01
BLENDER	blender	0.01	give	0.01	(0.01	around	0.01)	0.01
CUISINART	blender	0.03	dropped	0.02	holding	0.02	vase	0.02	out	0.01
CUTTINGBOARD	trash	0.01	pot	0.01	stuck	0.01	wall	0.01	best	0.01
REGISTER	bill	0.07	check	0.06	thank	0.02	no	0.02	pay	0.02
EMPTYTEACUP	tea	0.04	refill	0.01	:D	0.01	whenever	0.01	bon	0.01
EMPTYMUG	coffee	0.05	check	0.02	cup	0.01	thanks	0.01	.	0.01
EMPTYGLASS	beer	0.04	water	0.03	another	0.02	thanks	0.01	thirsty	0.01
EMPTYWINGLASS	wine	0.04	red	0.02	white	0.01	enjoy	0.01	glass	0.01
EMPTYBOWL	soup	0.03	finished	0.01	entree	0.01	yes	0.01	enjoy	0.01
EMPTYPLATE	enjoy	0.02	else	0.02	dessert	0.02	thank	0.02	anything	0.02
EMPTYWINEBOTTLE	happened	0.02	move	0.01	invisible	0.01	wall	0.01	wonderful	0.01
EMPTYWATERBOTTLE	bottle	0.04	they're	0.02	walk	0.01	vodka	0.01	cold	0.01
EMPTYFRUITBOWL	fruit	0.02	trash	0.01	serious	0.01	fish	0.01	lol	0.01
EMPTYCUTTINGBOARD	pot	0.01	lol	0.01	board	0.01	fish	0.01	best	0.01
EMPTYVASE	vase	0.04	flowers	0.03	flower	0.02	cost	0.02	they	0.02
EMPTYBRANDYBOTTLE	brandy	0.03	asl	0.02	alcoholic	0.02	whiskey	0.01	told	0.01
EMPTYTRASH	trash	0.04	woah	0.02	ew	0.02	garbage	0.02	flying	0.01
BAR	beer	0.21	water	0.15	wine	0.14	red	0.13	white	0.1
COUNTER	soup	0.06	steak	0.06	salad	0.05	lobster	0.05	tart	0.05
TABLE	please	0.06	water	0.05	wine	0.04	coffee	0.03	soup	0.03
CHAIR	seat	0.05	sit	0.04	table	0.04	anywhere	0.03	follow	0.03
STOOL	young	0.02	sup	0.01	wine	0.01	bar	0.01	boring	0.01
PODIUM	check	0.08	bill	0.07	else	0.02	no	0.02	the	0.02
MENUBOX	pleae	0.02	hold	0.01	dessert	0.01	second	0.01	minute	0.01
DISHWASHER	microwave	0.02	kitchen	0.01	theres	0.01	look	0.01	that	0.0
STOVE	w	0.01	k	0.01	its	0.01	know	0.00	in	0.00
FRIDGE	cost	0.01	staff	0.01	problems	0.01	vase	0.01	top	0.01
TRASHCOMPACTOR	of	0.01	wine	0.00	go	0.00	the	0.00	water	0.00
BARTENDER	bartender	0.03	excuse	0.01	alcoholic	0.01	doin	0.01	mine	0.01
CHEF	favor	0.02	trick	0.02	ha	0.02	ass	0.01	god	0.01

Table 1: all objects types and their 5 most strongly associated words in the preceding dialogue block

4 Finding words and phrases referring to items on the menu

We will now look in some more detail into the food-items that are ordered (including drinks), listed in table 2. In the present implementation we tell the system which item types to look at, but automatic object clustering does distinguish food and drink items, too (Orkin, 2007). These items are of interest for a number of reasons. Not only is it highly relevant for the performance of automated characters to be able to recognize which food-items are being talked about when, but they are also interesting because they can be referred to in various ways, and often by expressions consisting of more than one word. Furthermore, there are a number of relevant dialogue acts involving the words for these items, such as ordering. When we can identify the expressions referring to these

items, that will also help us identify the environments that these expressions occur in and their function or place in the game.

We will try to extract words and multi-word expressions referring to these objects. In order to avoid all suspicion that we are reproducing the scripted knowledge of the chef and the bartender, we take a slightly different road than before. The point where the customer orders an item is likely to occur earlier than in the dialogue block directly preceding the appearance of the item, or the moment he gets it. So if we want to bypass all interaction with the chef and bartender, it helps to make a rough assumption about where in the game the customer will order, rather than going by our general assumption above. Whereas the above assumption most likely applies to other domains too, this one is a specific assumption based on human knowledge of restaurant scenarios. We can-

not make it too specific though, because all games are different.

Every time the waitress puts down a food-item on a table², all customer utterances between this moment and the moment the customer first sat down in the game are considered context for this item. We will refer to this as the order-context for the item. The order-context for an item type is collected by joining the order-contexts of its instances. For the totals we add up the collected order-contexts of all items, rather than taking the totals of the whole corpus. This way we correct for anything that is counted double because it is part of the order-context of more than one item (order-contexts frequently overlap, as in most games more than one item is ordered). The size of this portion of the corpus, without the overlap, is 37,827 words.

4.1 Scoring words and multi-word sequences

Once more we find the most strongly associated words for each item, yielding results similar (but not identical) to table 1. We do the same for two-word and three-word sequences (bigrams and trigrams). For each item we accept the highest scoring word as a good word, assuming that in the minimal case an item can be referred to by exactly one single-word expression. To the extend that our method works, this expression should then be the one that scores highest. Next we accept bigrams that score above a certain threshold if their composing words also score above a threshold (We take $\phi > 0.02$ as a threshold for both). Words that occur in accepted bigrams, but had not been accepted yet, are added to the list of accepted words. Similarly, for trigrams we accept those that score high (same threshold used) and of which the composing bigrams have already been selected in the previous step.³ The found sequences are presented in table 2.

The approach is somewhat conservative, so we do miss some relevant words, such as ‘steak’ for FILET (which scored second among the words). We expect that we can catch these later by showing that they occur in the same environments as other food-item expressions. Similarly for the

²We could make sure that it is the table the customer actually sits at, but since we only have one customer, the extra administration this would require would probably come with very little gain.

³Looking at four-word sequences does not yield additional results if we require that their components have to have been already accepted.

item type	unigrams	bigrams	trigrams
WATER	‘water’	-	-
TEA	‘tea’	-	-
COFFEE	‘coffee’	-	-
BEER	‘beer’	-	-
REDWINE	‘red’ ‘wine’	‘red wine’	-
WHITEWINE	‘white’ ‘wine’	‘white wine’	-
SOUP	‘soup’ ‘du’ ‘jour’ ‘vegetable’	‘soup du’ ‘du jour’ ‘vegetable soup’	‘soup du jour’
SALAD	‘salad’ ‘cobb’	‘cobb salad’	-
SPAGHETTI	‘spaghetti’ ‘marinara’	‘spaghetti marinara’	-
FILET	‘filet’ ‘mignon’	‘filet mignon’	-
SALMON	‘salmon’ ‘grilled’	‘grilled salmon’	-
LOBSTER	‘lobster’ ‘thermador’	‘lobster thermador’	-
CHEESECAKE	‘cheesecake’ ‘cherry’	‘cherry cheesecake’	-
PIE	‘pie’ ‘berry’	‘berry pie’	-
TART	‘tart’ ‘nectarine’	‘nectarine tart’	-

Table 2: extracted words, bigrams, and trigrams for the items on the menu

more general words ‘fish’ and ‘pasta’ for SALMON and SPAGHETTI respectively, that we saw in table 1. These additionally turn out to have a less strong presence in this part of the data. Presumably they are not used that much in ordering, perhaps because customers, in this situation, tend to repeat what they read on the menu.

4.2 Filtering referring expressions

We now have identified words and sequences that can be involved in referring to food-items, but we still don’t know which of these can be used by themselves for this purpose, and which only as part of a longer sequence. What we do next is to score all words and the selected bigrams and trigrams together in such a way that we only count bigrams where they are not part of one of the selected trigrams and only count the words where they are not part of any of the selected bigrams or trigrams. That is, we treat the bigrams and trigrams selected in the previous step as words, and ignore their internal structure, so we can compare the association scores of these ‘words with spaces’ to those of other words and in particular with those of their composing words in other configurations. The selected words and bigrams that still score above the threshold now, can apparently refer independently to their associated food-items. This give us the results shown in table 3.

item type	referring expressions
WATER	'water'
TEA	'tea'
COFFEE	'coffee'
BEER	'beer'
REDWINE	'red' 'wine' 'red wine'
WHITEWINE	'white' 'white wine'
SOUP	'soup' 'jour' 'vegetable soup' 'soup du jour'
SALAD	'salad' 'cobb salad'
SPAGHETTI	'spaghetti' 'spaghetti marinara'
FILET	'filet' 'filet mignon'
SALMON	'salmon' 'grilled salmon'
LOBSTER	'lobster' 'lobster thermador'
CHEESECAKE	'cheesecake' 'cherry cheesecake'
PIE	'pie' 'berry pie'
TART	'tart' 'nectarine tart'

Table 3: extracted referring expressions for the items on the menu

There are two things in this table that are counter-intuitive. Firstly, on the precision side, 'jour' appears to be used outside the expression 'soup du jour' to refer to SOUP, which is quite odd. The most likely cause is that 'du' is relatively often written as 'de', although just not often enough for the whole alternative construction to be picked up (16 times on a total of 79). This issue can be resolved by applying spelling normalization, to recognize that the same word can have different written forms, which will be important to make the final system interact robustly, in any case. As expected in a chat set-up, the spelling is overall rather variable. The opportunities for spelling normalization, however, are promising, since we do not only have linguistic context but also non-linguistic context to make use of. Nevertheless, the theme falls beyond the scope of this paper.

Secondly, on the recall side, 'wine' does not show up as a word that can independently refer to WHITEWINE. Actually, the whole wine situation is a bit particular. Because the word 'wine' occurs prominently in the context of both WHITEWINE and REDWINE it doesn't associate as strongly with either of them as the words 'red' and 'white', which distinguish between the two. In the present implementation our algorithm is not aware of similarities between the two types of ob-



Figure 2: menu and specials board from the Restaurant Game

jects, which could provide support for the idea that 'wine' is used with the same meaning in both cases. Recognizing and using such similarities remains for future work. It may not seem straightforward either that 'red' and 'white' can refer independently to their objects. What happens is that in the data the word 'wine' can easily occur in a previous utterance of either the customer or the waitress, e.g. waitress: '*would you like some wine?*', customer: '*yes, red, please.*'. Whether this can be called independent reference is questionable, but at its present level of sophistication, we expect our extraction method to behave this way. Also, because of the medium of chat, players may tend to keep their utterances shorter than they would when talking, using only the distinctive term, when it is clear from the context what they are talking about⁴. Also 'house red/white (wine)' patterns (as appear on the menu in figure 2) do occur in the data but our method is not sensitive enough to pick them up.⁵

In spite of the imperfections mentioned in this step and the previous one (mainly recall issues), we will see in the next section that the expressions we extracted do give us a good handle on extracting patterns of ordering food.

⁴Note that our hard-coded bartender does not respond to the ambiguous order of 'wine' either, as the human designer had the same intuition, that 'red' and 'white' are more reliable.

⁵We are not concerned about not retrieving 'glass of' construction, because we consider it not to be part of the core referring expressions, but a more general construction that applies to all cold drinks.

5 How to order

Now that we have a close to comprehensive collection of expressions referring to food-items, we will use these to find the ‘constructions’ used for ordering these. For each food-item being put on the table, we record the most recent utterance that contains one of its corresponding referring expressions. We replace this expression by the placeholder ‘<FoodItem>’, so that we can abstract away from the particular expression or its referent, and focus on the rest of the utterance to find patterns for ordering. Table 4 presents the utterance patterns that occurred more than once in a condensed way.⁶

<p>(and) (a/the/one/another/a glass of/more/some/my) <FoodItem> (and) (.) (please/plz) (.) (a/the) <FoodItem></p> <p>just (a/some) <FoodItem> please yes (.) (a) <FoodItem> (please) (and a) <FoodItem> ?</p> <p>glass of/with a/2/um/then/sure <FoodItem> <FoodItem> 2/too/to start! <FoodItem> is fine a <FoodItem> would be great</p> <p>where is my <FoodItem> steak and <FoodItem></p> <p>i want <FoodItem> (and a <FoodItem>)</p> <p>i'd/i would like (to start with/to have) (a/the/some/a glass of) <FoodItem> (please) (.) i will like <FoodItem></p> <p>i will start with <FoodItem> i'll take a <FoodItem></p> <p>(i think/believe) i'll/i will/ill have (the/a) <FoodItem> (and) (please) (.) a glass of <FoodItem></p> <p>can/could i have/get (a/the/some/some more/a glass of) <FoodItem> (and) (please) (?) (a) <FoodItem></p> <p>may i have a/the/some <FoodItem> (please) (?) may i please have a glass of <FoodItem> ? please may i have the <FoodItem> ?</p>
--

Table 4: condensed representation of order-utterances found more than once

There are 492 utterance patterns that occurred more than once, plus another 1195 that occurred only once. Those that occurred twice or more are basically all reasonable ways of ordering food in a restaurant (although some might be the renewal of an order rather than the original one). The vast majority of the patterns that occurred only once were also perfectly acceptable ways of ordering. Many had substantial overlap with the more frequent patterns, some were a bit more original or contained extra comments like ‘*i'm very hungry*’. We can

⁶It is worth noting that 97 of the utterances consisted only of ‘<FoodItem>’. They are included in the first generalized pattern.

conclude that there is a lot of variation and that here the extraction method shows a real potential of outperforming hand-coding. As for recall, we can be sure that there are patterns we missed, but also that there will be many possible patterns that do simply not occur in the data. To what extend we will be able to recognize food orders in future games, will largely depend on how successfully we can generalize over the patterns we found.

We envision encoding the extracted linguistic knowledge in the form of a construction grammar (e.g. (Croft, 2001)). The extracted patterns could already be used as very coarse grained constructions, in which <FoodItem> is a slot to be filled by another construction.⁷ At the same time it is clear that there are many recurrent patterns in the data that could be analyzed in more detail. We show initial examples in the subsections 5.1 and 5.2. As for meaning, at utterance level, an important aspect of meaning is the utterance’s function as a dialogue act. Rather than describing what happens, most utterances in this game are part of what happens in a similar way as the physical actions are (Searle, 1965). Knowing that something is being ordered, what is being ordered, and how ordering acts fit into the overall scenario will be extremely useful to a planner that drives AI characters.

5.1 Identifying coordination

If we look at sequences associated with ordering, we see that many of them contain more than one <FoodItem> expression. These tend to be separated by the word ‘and’. We can support this observation by checking which words are most strongly associated with order phrases that contain 2 or more instances of ‘<FoodItem>’. The 10 most strongly associated words and their scores are: ‘and’(0.19), ‘de’(0.05), ‘,’(0.04), ‘minon’(0.04), ‘i’ll’(0.03), ‘&’(0.03), ‘dessert’(0.02), ‘the’(0.02), ‘with’(0.02), ‘n’(0.02). The word ‘and’ comes out as a clear winner.

Of course ‘coordination’ is a more general concept than is supported by the data at this point. What is supported is that ‘and’ is a word that is used to squeeze two <FoodItem> expressions into a single order.

⁷Lieven et. al. (2003) argue that young children continue to rely on combining just two or three units well beyond the two-word stage.

5.2 Class-specific constructions

Some of the variation we saw in table 4 is related to there being different types and classes of items that can be distinguished. Table 5 shows this for some relevant classes. This can help us extract more local constructions within the order-phrases and tell the difference between them.

class	trigram	phi
COLDDRINK	'a glass of'	0.06
	'glass of <FoodItem>'	0.06
	'glasses of <FoodItem>'	0.05
HOTDRINK	'cup of <FoodItem>'	0.07
	'a cup of'	0.06
	'<FoodItem> and <FoodItem>'	0.03
STARTER	'<FoodItem> and <FoodItem>'	0.05
	'a <FoodItem> and'	0.04
	'<FoodItem> to start'	0.04
ENTREE	'have the <FoodItem>'	0.07
	'the <FoodItem> and'	0.05
	'and the <FoodItem>'	0.05
DESSERT	'<FoodItem> for dessert'	0.04
	'piece of <FoodItem>'	0.04
	'a piece of'	0.03

Table 5: some interesting classes of item types and their most strongly associated trigrams with phi scores

Here we hand-assigned classes and showed the differences in language, but we could of course start from the other end, and automatically cluster item types on the basis of how they are ordered, thus creating the classes.

6 Finding more words for food-items

It would be great if we could use the knowledge about what ordering looks like, to identify situations where the customer orders something that is not on the menu and figure out how to respond to that (– a challenge because of sparse data).

We extracted the 30 environments of '<FoodItem>', consisting of 2 words to the left and 2 words to the right, that were most strongly associated with ordering, and counted what other words occurred in these environments in the ordering parts of the games. These were the words found, with the number of times they were found in these environments:

'yes'(69), 'menu'(27), 'steak'(16), 'check'(5), 'bill'(5), 'coffe'(4), 'spaghetti'(3), 'desert'(3), 'dinner'(3), 'cofee'(2), 'seat'(2), 'fillet'(2), 'sit'(2), 'more'(2), 'dessert'(2), 'you'(2), 'no'(2), 'coke'(2), 'drink'(2), 'bear'(2), 'cute'(1), 'vest'(1), 'help'(1), 'cheese'(1), 'sweet'(1), 'fish'(1), 'ea'(1), 'glass'(1), 'sphagetti'(1), 'burger'(1), 'manager'(1), 'mignon'(1), 'chat'(1), 'cutlery'(1), 'iyes'(1), 'one'(1), 'tab'(1), 'bathroom'(1), 'sieve'(1), 'chesscake'(1), 'selmon'(1), 'med'(1), 'question'(1), 'fast'(1), 'redwine'(1),

'bees'(1), 'bread'(1), 'pudding'(1), 'trash'(1), '?'(1), 'pizza'(1), 'fight'(1), 'cheescake'(1), 'wime'(1), 'wate'(1), 'grilled'(1), 'moment'(1), 'beeer'(1), 'here'(1), '...' (1), 'spegetti'(1), 'pasta'(1), 'spagattie'(1), 'win'(1), 'thank'(1), 'cold'(1), 'main'(1), 'broiler'(1), 'marinara'(1), 'u'(1), 'h'(1), 'refill'(1), 'brandy'(1), 'um'(1), 'whiskey'(1), 'meni'(1), 'acoke'(1), 'cake'(1), 'soda'(1), 'fun'(1), 'offe'(1), 'scotch'(1), 'yours'(1)

These first results look promising and it should not be too hard to filter out misspellings of known words, alternative ways of referring to known food-items, and words that clearly refer to something else known (such as 'menu') (or are simply so frequent that they just have to have some other function). Still, we conclude that the present method on 1000 games is not yet sensitive enough to confidently pick out other food-terms. Improving it remains for future work. This is, on the other hand, a good point to recuperate expressions such as 'steak', which we missed earlier.

7 Discussion/Conclusion

We have picked up all of the menu descriptions for the food-items plus most of the sensible shorter forms. This was good enough to identify patterns of how to order these items.

Our extraction methods have so far been rather human-guided. It would be interesting to see if it is possible to design a more generalized procedure that automatically generates hypotheses about where to look for associations, and what assumptions about the workings of natural language it needs to be equipped with. One basic thing we have used in this case, is the idea that linguistic expressions can be used to refer to things in the non-linguistic context. Another one that is very relevant in The Restaurant Game is that utterances are used as dialogue acts, with very strong parallels to physical actions.

We hope to have given an impression of the richness of this dataset and the possibilities it offers. We argue that finding referring expressions for concrete objects in a simple way is a good starting point in this kind of data to get a handle on more abstract constructions, too.

Acknowledgments

This research was funded by a Rubicon grant from the Netherlands Organisation for Scientific Research (NWO), project nr. 446-09-011.

References

- W. Croft. 2001. *Radical Construction Grammar*. Oxford University Press, New York.
- H.T. Dang, D. Kelly, and J. Lin. 2007. Overview of the TREC 2007 Question Answering Track. In EM Voorhees and Lori Buckland, editors, *The Sixteenth Text REtrieval Conference Proceedings 2007*, number 500-274 in Special Publication, Gaithersburg, Maryland. NIST.
- M. Fleischman and D. Roy. 2005. Why verbs are harder to learn than nouns: Initial insights from a computational model of intention recognition in situated word learning. In *27th Annual Meeting of the Cognitive Science Society, Stresa, Italy*.
- D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. 2007. The third PASCAL Recognizing Textual Entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Rochester, New York. Association for Computational Linguistics.
- P. Gorniak and D. Roy. 2004. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21(1):429–470.
- P. Gorniak and D. Roy. 2005. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *Proceedings of the 7th international conference on Multimodal interfaces*, page 143. ACM.
- D. Hewlett, S. Hoversten, W. Kerr, P. Cohen, and Y.H. Chang. 2007. Wubble world. In *Proceedings of the 3rd Conference on Artificial Intelligence and Interactive Entertainment*.
- S. Kopp, B. Jung, N. Leßmann, and I. Wachsmuth. 2003. Max - A Multimodal Assistant in Virtual Reality Construction. *KI*, 17(4):11.
- E. Lieven, H. Behrens, J. Speares, and M. Tomasello. 2003. Early syntactic creativity: A usage-based approach. *Journal of Child Language*, 30(02):333–370.
- C.D. Manning and H. Schütze. 2000. *Foundations of statistical natural language processing*. MIT Press.
- J. Orkin and D. Roy. 2007. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development*, 3(1):39–60.
- J. Orkin and D. Roy. 2009. Automatic learning and generation of social behavior from collective human gameplay. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 385–392. International Foundation for Autonomous Agents and Multiagent Systems.
- J. Orkin. 2007. Learning plan networks in conversational video games. Master’s thesis, Massachusetts Institute of Technology.
- D. Roy. 2005. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1-2):170–205.
- R.C. Schank and R.P. Abelson. 1977. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum Associates Hillsdale, NJ.
- J. Searle. 1965. What is a speech act? *Perspectives in the philosophy of language: a concise anthology*, 2000:253–268.
- L. Steels. 2003. Evolving grounded communication for robots. *Trends in cognitive sciences*, 7(7):308–312.
- A. Stefanowitsch and ST Gries. 2003. Collostructions: Investigating the interaction of words and constructions. *International Journal of Corpus Linguistics*, 8(2):209–243.
- L. Von Ahn and L. Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM.

EXCERPT – Ein integriertes System zum informativen Summarizing und Within-Document-Retrieval

Jürgen Reischer

Informationswissenschaft

Universität Regensburg

Regensburg, Deutschland

juergen.reischer@sprachlit.uni-regensburg.de

Abstract

Das experimentelle EXCERPT-System wird vorgestellt, das eine Suchmaschine zur Auswahl und Ausgabe von Information innerhalb von Textdokumenten realisiert. Die Techniken des informativen Summarizings werden auf ein System zum Within-Document-Retrieval übertragen, das einen integrierten Ansatz zur suchbegriffs-basierten Extraktion relevanter Textinhalte und deren nutzergerechte Aufbereitung zu informativ-zusammenfassenden Ergebnis-Einheiten bietet.

1 Einleitung

Angesichts des Umfangs textueller Information, die uns täglich in elektronischer Form erreicht, sind Systeme zur gezielten Suche und Auswahl von Information innerhalb von Dokumenten nach den Bedürfnissen des Nutzers unverzichtbar. Umso erstaunlicher erscheint es daher, dass gerade Systeme wie WWW-Browser, PDF-Reader und selbst Textverarbeitungen meist nur über einfachste Verfahren des Zusammenfassens oder Suchens von Information verfügen. Kann für Ersteres noch das ein oder andere Plugin installiert werden, so muss sich der Nutzer im letzteren Falle mit rudimentären Suchen-Kommandos zufrieden geben, die zumeist nicht einmal die Eingabe mehrerer Suchbegriffe erlauben (die Google-Buch-Suche stellt dabei eine Ausnahme dar). Bei der Suche nach benötigter Information ist der Nutzer auch bei relativ kurzen Dokumenten letztlich nicht gewillt, den gesamten Text intellektuell auf die relevanten Inhalte hin zu durchsuchen. Mit der zunehmenden Digitalisierung immer umfangreicherer Dokumente wird sich dieses Problem weiter verschärfen. Die Aufgabe zur Lösung des Problems kann nur darin bestehen, maschinelle Verfahren und Systeme zu entwickeln, die den Nutzer bei der inhaltlichen Arbeit mit Texten unterstützen.

Das hier vorgestellte experimentelle EXCERPT-

System¹ zeigt einen Ansatz auf, der bei der Suche, Auswahl und Ausgabe von Information aus (englischen) Dokumenten einen neuen Weg beschreitet: Der Nutzer kann durch Angabe ein oder mehrerer Suchbegriffe thematisch relevante Textpassagen (Sätze/Absätze) aus Dokumenten ermitteln, die dann nach den Vorgaben des Nutzers zu einer 'gerankten' Liste von informativen Summarys aus ein oder mehreren (Ab-)Sätzen zusammengefasst werden. Dies erlaubt dem Nutzer einen schnellen Überblick über die ihn interessierenden Inhalte, die er inkrementell rezipieren kann, bis sein Informationsbedarf gedeckt ist. In diesem Sinne stellt das EXCERPT-System eine Art Suchmaschine für Textdokumente dar, die Textpassagen anhand von Suchbegriffen ermittelt und als summaryartige 'Snippets' in einer gerankten Ergebnisliste ausgibt. Real existierende Systeme zum Within-Document-Retrieval sind rar: Hier ist vor allem das ProfileSkim-System von Harper et al. (2004) zu nennen, das die suchbegriffs-basierte Bewertung und Selektion von Textpassagen unterstützt; auch das TOP[OGRAPH]IC-System von Hahn und Reimer (1986) bietet eine vergleichbare Funktionalität. Andere Arten von Passagen-Retrieval-Systemen erlauben dem Nutzer hingegen keinen direkten Zugriff auf die Textinhalte und dienen hauptsächlich dem übergeordneten Dokumenten-Retrieval (zur Unterscheidung verschiedener Systemtypen vgl. Stock, 2007). Im Gegensatz hierzu stehen zahlreiche Summarizing-Systeme mit unterschiedlichen Zielsetzungen (vgl. die aktuelle Bibliografie in Radov, 2009). Am ehesten vergleichbar mit Within-Document-Retrieval-Systemen sind nutzer- oder topik-zentrierte Summarys, die nach Angabe von Thematemen aus einem oder mehreren Dokumenten ein entsprechendes Summary zusammenstellen, jedoch keine Suchmaschinen-Funktionalität anbieten.

¹EXpert in Computational Evaluation and Retrieval of Passages of Text.

2 Konzeption und Realisierung

EXCERPT als integriertes Summarizing- und Passagen-Retrieval-System legt den Fokus auf die Suche und Extraktion informativer Textausschnitte im Sinne des informativen Summarizings und Within-Document-Retrievals.² Der Nutzer kann mittels Suchbegriffen interessierende Textpassagen ermitteln, die nach Relevanz und Informativität sortiert ausgegeben werden.³ Konzeptuell lassen sich sowohl Summarizing als auch Passagen-Retrieval als Formen der Informationsselektion begreifen, wobei Information aus Texten nach bestimmten Kriterien selektiert wird: Bei der Generierung von extraktiven Summarys werden wichtige (thematisch informative, interessante, repräsentative) Sätze oder Absätze selektiert, beim Retrieval von Information relevante (thematisch auf die Suchbegriffe passende) Passagen. Summarizing wird dabei letztlich als Sonderfall des Within-Document-Retrievals bei der Ermittlung und Extraktion relevanter Information betrachtet: Gibt der Nutzer keine thematisch einschränkenden Suchbegriffe an, werden alle Sätze/Absätze des Textes als gleich relevant erachtet und die Passagen ausschließlich gemäß ihrer Wichtigkeit beurteilt (analog einem generischen, informativ-extraktiven Summary).

Bei Angabe von einem oder mehreren Suchbegriffen werden die Passagen zusätzlich hinsichtlich ihrer Relevanz bezüglich der Suchbegriffe bewertet und selektiert. Die Umsetzung dieses Ansatzes in ein lauffähiges System erfordert dabei die Realisierung zweier wesentlicher Komponenten: (i) ein Analysemodul, das den Text formal und inhaltlich im Hinblick auf Summarizing- und Within-Document-Retrieval-Aufgaben analysiert; (ii) ein Retrieval- und Ranking-Modul, das auf Basis des analysierten Textes die Suchanfragen des Nutzers bedient.

2.1 Formale und inhaltliche Textanalyse

Die Analyse eines Textes erfolgt formal und inhaltlich: Die formale Analyse umfasst vor allem die Zerlegung des Textes in relevante Texteinheiten und deren Normalisierung, die inhaltliche Analyse bezieht sich auf die semantischen Eigenschaf-

²Zum Gegensatz indikative vs. informative Summarys vgl. Borko und Bernier (1975).

³Auf eine Diskussion des Informations- und Informativitätsbegriffs muss hier aus Platzgründen verzichtet werden; zu Ersterem siehe Reischer (2006), zu Letzterem z. B. de Beaugrande und Dressler (1981).

ten der Einheiten und deren thematische Zusammenhänge im Text.

Bei der formalen Analyse des Textes wird in mehreren Schritten eine Strukturerkennung und Normalisierung durchgeführt: Bei der Vorverarbeitung werden Quotationszeichen und Parenthesen vereinheitlicht (z. B. “-” zu “-”), mehrfache Satzende-Zeichen reduziert (z. B. “??” zu “?”) sowie bekannte Abkürzungen aufgelöst (z. B. “Brit.” zu “british” expandiert, um die nachfolgende Satzerkennung zu vereinfachen). Darauf aufbauend können im nächsten Schritt Absätze anhand von Leerzeilen bzw. “Carriage Return” und/oder “Line Feed” erkannt werden. Der erkannte Absatz wird anhand von Whitespace-Zeichen in Tokens zerlegt, um anschließend die Satzerkennung durchführen zu können: Als Satzende-Zeichen werden “?”, “!” und “.” gewertet, nicht aber “;” und “:”. Für jeden Satz werden sämtliche Inhaltsterme normalisiert, wobei das WordNet-Lexikon zum Einsatz kommt (Fellbaum, 1998): Regelmäßige Terme werden anhand einer Liste regulärer Flexionsmuster deflektiert, unregelmäßige Terme über eine entsprechende Liste normalisiert. Alle in Bezug auf das WordNet-Lexikon unbekannten Terme werden als Komposita (ohne Leerzeichen) interpretiert und in ihre Bestandteile zerlegt; sind auch die Bestandteile unbekannt, wird der Term als Neologismus gewertet. Nach der Normalisierung der Satzterme wird innerhalb jedes Satzes eine Mehrwortterm-Erkennung durchgeführt, die bekannte mehrteilige lexikalische Ausdrücke (separiert durch Leerzeichen) mit Hilfe des WordNet-Lexikons zu einem einzigen Term zusammenfasst (z. B. “President” + “of” + “the” + “United” + “States” zu “President of the United States”).⁴

Die erkannten Lemmata werden in einem Termindex verwaltet, der die Zuordnung jedes Terms zu allen Sätzen, in denen er auftritt, enthält. Funktionswörter werden nicht weiter betrachtet, jedoch in einem eigenen Index gespeichert, der für die Ermittlung der Informativität einer Texteinheit benötigt wird. Gänzlich ausselektiert werden so genannte ‘Void words’ wie “Mr.” oder “alas” usw., die aufgrund ihres beschränkten semantischen Gehalts vergleichbar hochfrequenten Inhaltswörtern nicht weiter verarbeitet werden.

Für die semantisch-thematische Analyse des Textes wird die inhaltliche Struktur einzelner Ter-

⁴Aus Platzgründen kann hier auf die damit verbundenen Probleme und den Erkennungsalgorithmus nicht eingegangen werden (vgl. Reischer, 2010).

me oder Sätze im Hinblick auf deren Informativität untersucht. Die ermittelten Ergebnisse dienen später als Parameter bei der Berechnung von Informativitäts-Bewertungen ('Scores') für Sätze und Absätze. Der semantische Wert oder Informationsgehalt von Termen und Sätzen lässt sich indirekt anhand verschiedener Oberflächen-Merkmale des Textes feststellen, die sich allesamt maschinell ermitteln lassen. Die folgenden Parameter wurden zum Großteil aufgrund von Erkenntnissen der Literatur zum Summarizing ermittelt (Edmundson, 1969; Paice, 1990; Mittal et al., 1999; Goldstein et al., 1999; Banko et al., 1999; Hovy, 2004), ergänzt um linguistische Auswertungen eigener Korpora zur Informativität von Texten:⁵

- Spezifität: Der Informationsgehalt von Termen und Sätzen kann an deren semantischer Spezifität festgemacht werden. Bei Termen dient zum einen ihre Frequenz im Text vs. der Sprache als Indikator für ihren semantischen Gehalt: Terme, die im Text häufig, in der Sprache (bzw. einem repräsentativen Referenzkorpus) hingegen selten auftreten, werden als informativer betrachtet (vgl. entsprechend das tf•idf-Maß des Information-Retrievals; Salton und Buckley, 1988). Neologismen sind in dieser Hinsicht semantisch am gehaltvollsten, da sie in der Sprache bislang nicht aufgetreten sind. Individuausdrücke (Eigennamen) können unabhängig von ihrer Frequenz als semantisch so spezifisch betrachtet werden, dass sie nurmehr genau eine Entität bezeichnen (im Gegensatz zu Allgemeinbegriffen); zudem wird Eigennamen ein besonderer Interessantheitswert zugesprochen (Flesch, 1948).
- Semantische Intensität: Steigerungsformen von Adjektiven und partiell Adverbien werden als Indikator für semantisch aussagekräftige Sätze verwendet (Mittal et al., 1999; Goldstein et al., 1999). Vergleiche durch Komparative und Wertungen durch Superlative weisen auf relevante Inhalte oder Erkenntnisse hin, die für den Rezipienten von Interesse sein können ("X is better than Y", "the best X is Y"). Im Gegensatz hierzu zeigen Terme wie "by the way" oder "alas" Sätze an,

⁵Viele der genannten Parameter wurden durch die eigene Auswertung von Texten bestätigt. Dies war insofern nicht zu erwarten, als bisherige Untersuchungen nicht explizit auf die Frage der Informativität von Sätzen bzw. Summarys abzielten. Für eine detaillierte Darstellung aller Auswertungen muss auf Reischer (2010) verwiesen werden.

die nur nebensächliche Information vermitteln wollen. In diesem Sinne stellen solche Ausdrücke 'Malus'-Terme dar, wohingegen Steigerungsformen als Bonusterme betrachtet werden. Beide Sorten von Termen stehen in einer vordefinierten und nutzererweiterbaren Liste zur Verfügung.

- Informationalität: Informative Sätze sind vor allem Aussagesätze, wohingegen Fragesätze keine Information liefern, sondern gerade erfragen.⁶ Die Auswertung eines eigenen Korpus zur Frage der Informativität (vgl. Abschnitt 3) ergab, dass in der Menge der von den Testpersonen als informativ beurteilten Sätze kein einziger Frage- oder Befehlssatz enthalten war; d. h. solche Sätze sind für informative Summarys nicht geeignet (vgl. Alfonseca und Rodríguez, 2003). Zusätzlich wurden vor allem Aussagesätze innerhalb von Aufzählungen als überdurchschnittlich informativ gewertet. Als weiterer Faktor für Informationalität wird der Anteil an Pronomen der 1. Person ermittelt: Sätze mit 1.-Person-Pronomen werden offenbar als eher meinungs- statt faktenorientiert wahrgenommen. Die Auswertung des genannten Korpus ergab eine deutlich geringere Anzahl von Pronomen der 1. Person bei Sätzen, die als informativ beurteilt wurden.
- Topikalität: Thematisch zentrale Sätze geben den informationellen Kern eines Textes wieder. Die Zentralität einer Aussage lässt sich zum einen durch die inhaltliche Relationiertheit der Terme eines Satzes ermitteln: Je mehr semantisch-lexikalische Relationen ein Term aufweist, desto zentraler ist er für den Inhalt eines Textes. Der Grad der Relationiertheit von Termen und Sätzen wird über das WordNet-Lexikon berechnet (vgl. den verwandten Ansatz zu lexikalisch-thematischen Ketten z. B. in Barzilay und Elhadad, 1999). Zum anderen wird die topikalische Relevanz eines Terms durch seine Stellung im Satz ermittelt: Terme (vor allem Nomen) in syntaktischer Subjekts- oder Topikposition werden als thematisch zentraler erachtet als Terme in anderen Positionen (Subjekt-Prädikat- bzw. Topic-Comment-

⁶Rhetorische Fragen könnten einen Informationswert besitzen; diese treten jedoch eher selten auf und sind formal kaum von nicht-rhetorischen Fragen zu unterscheiden.

Struktur). Tendenziell befindet sich das Subjekt/Topik eines Satzes in der ersten Hälfte der Satzwörter, so dass die Position eines Terms im Satz als Anhaltspunkt hierfür verwendet werden kann.

- Positionalität: Sätze zu Beginn eines Absatzes oder Abschnitts bzw. unmittelbar unterhalb einer Überschrift leiten in der Regel ein neues Thema ein und sind daher als inhaltlich signifikant einzustufen (Baxendale, 1958; Edmundson, 1969).
- Innovativität: Als Indikator für nicht-redundante (informative) Aussagen kann die Anzahl erstmals (nicht erneut) auftretender Terme in einem Satz/Absatz herangezogen werden (von Weizsäcker, 1974). Zudem deuten auch hier wieder Neologismen auf spezifische neue Konzepte hin, die für den Rezipienten von Interesse sein können (Elsen, 2004).

Aus Platzgründen kann hier nur grob auf den Algorithmus zur Verrechnung der Parameter eingegangen werden: Die ermittelten Parameter werden zu sieben Kategorien zusammengefasst, für die pro (Ab-)Satz des Textes jeweils ein eigenes Ranking (bezüglich der Parameter der Kategorie) durchgeführt wird. Die Ranking-Positionen aller Kategorien werden für jeden (Ab-)Satz gemittelt und als Gesamt-Rankingwert betrachtet. Entsprechend diesem Wert werden die (Ab-)Sätze zu möglichst informativen Summarys zusammengefasst und ausgegeben (unter Beachtung ihrer thematischen Relevanz, sofern Suchterme vorgegeben wurden).

2.2 Retrieval und Ranking von Textpassagen

Die konzeptionelle Integration von Passagen-Retrieval (im Sinne des Within-Document-Retrievals) und Summarizings findet seine Realisierung in der Art und Weise, wie der Nutzer Information selektieren kann: Gibt er 0 Suchbegriffe an, werden alle Einheiten des Textes grundsätzlich als relevant erachtet, da der Nutzer keine thematische Einschränkung vorgegeben hat; gibt er hingegen 1 bis N Suchterme an, werden nur die entsprechend thematisch relevanten Einheiten aus dem Text selektiert. In beiden Fällen werden alle selektierten Einheiten mit ihren Relevanzwerten in einer Liste gespeichert, von wo aus sie zu größeren Ausgabeclustern (Summarys) zusammengefasst und gemäß ihren Relevanz-

und Informativitäts-Bewertungen ausgegeben werden. Zu diesem Zweck kann der Nutzer darüber bestimmen, wie viele Einheiten im Sinne einzelner Sätze oder Absätze zu einer Ausgabeeinheit ('Cluster') zusammengefasst werden sollen. Dies entspricht in etwa informativen, summary-artigen Text-'Snippets' aus jeweils 1 bis M zuvor selektierten (Ab-)Sätzen, die zu Adhoc-Clustern verbunden und ausgegeben werden. Die Suchmaschine für Texte zeigt damit nicht nur wie gewohnt die Fundstellen an, sondern fasst die gefundenen Ergebnisse auf Wunsch zu Summarys bestimmter Größe zusammen und gibt diese nach Relevanz und Informativität sortiert aus (vgl. beispielhaft Abb. 1 in Anhang A).

Der Sortierungsprozess für das Ranking berücksichtigt sowohl die Relevanz- als auch Informativitätswerte der erzeugten Ausgabecluster. Diese kann man zum einen als nutzer- bzw. topikzentrierte informative Summarys verstehen, falls thematisch beschränkende Suchterme vorgegeben wurden; zum anderen handelt es sich um generische informative Summarys, wenn durch den Nutzer kein Thematerm angegeben wurde.⁷ Im ersten Schritt werden die Summarys dabei nach Relevanz hinsichtlich der Suchbegriffe sortiert; hat der Nutzer keine Suchterme angegeben, weil er generische Summarys wünscht, werden alle Summarys als gleich relevant erachtet. Im zweiten Schritt erfolgt innerhalb gleich relevanter Summarys die Sortierung gemäß ihrer Informativität, die zuvor bereits in der Analysephase für einzelne Sätze und Absätze ermittelt wurde. Primär werden die Ausgabecluster im Sinne von Summarys also nach Relevanz, sekundär nach Informativität gerankt.

Dem Nutzer steht hierbei eine Reihe von Möglichkeiten zur Verfügung, wie er auf die Suche nach Inhalten und deren Ausgabe Einfluss nehmen kann. Im Hinblick auf die Retrieval-Funktionalität kann sich der Nutzer zwischen Precision und Recall entscheiden: Sollen die gesuchten Begriffe exakt innerhalb einer Passage gefunden werden oder können die Passagen auch synonyme oder semantisch relationierte Terme enthalten (z. B. Hyp[er]onyme und Antonyme)? Im letzteren Falle werden entsprechend ermittelte Passagen umso niedriger im Hinblick auf das Ranking bewertet, je größer die semantische Distanz zwischen

⁷Im einen Grenzfall bestehen die Summary-Cluster jeweils nur aus einem einzigen Satz oder Absatz, im anderen Grenzfall aus dem gesamten Text.

Such- und Textterm ist. Im Grenzfall treten die Suchbegriffe im Text überhaupt nicht auf, sondern die gesuchten Konzepte werden über Synonyme oder thematisch relationierte Terme ermittelt (vage/thematische Suche). Im Hinblick auf das Ranking der konstruierten Ausgabe-Cluster (Summarys) stehen dem Nutzer verschiedene Optionen zur Wahl: Die konstruierten Ausgabecluster können aus einer oder mehreren, kontinuierlich oder (dis)kontinuierlich im Text aufeinander folgenden Einheiten bestehen. Wird pro Ausgabecluster nur jeweils genau ein Satz oder Absatz ausgegeben, entspricht dies der Ausgabegröße herkömmlicher Dokumenten-Suchmaschinen, die jeweils genau ein Treffer-Item pro Rangplatz ausgeben.⁸

Werden die Einheiten im Ausgabecluster in kontinuierlicher Originalreihenfolge präsentiert, kann die Lesbarkeit und Kohärenz erhöht werden; die beste Lesbarkeit ergibt sich dabei, wenn anstelle von Sätzen ganze Absätze zu Ausgabeclustern zusammengefügt werden. Die inhaltliche Kohärenz ist generell durch die thematisch selektive Extraktion von (Ab-)Sätzen gewährleistet. Die formale Kohäsion hingegen kann kaum verbessert werden, da dies einen Eingriff in die Originalstruktur der Sätze oder Absätze bedeuten würde (z. B. durch Auflösung von Pronomen und Abkürzungen, Entfernung konjunktional gebrauchter Adverbien). Vom Nutzer könnte dies jedoch als Manipulation des Originaltextes missverstanden werden, zumal er durch den Umgang mit Ergebnissen von (Web-)Suchmaschinen gewohnt ist, dass die gefundenen Dokumente im Originalzustand präsentiert werden (ansonsten entsteht womöglich der Eindruck von Zensur).

3 Evaluation

Um die Leistung des informativitäts-basierten Ansatzes zum automatischen Summarizing und Within-Document-Retrieval zu testen, wurde eine Evaluation durchgeführt. Hierfür wurden zum einen zwei frei verfügbare Summarizing-Korpora von Zechner (1995) und Hasler et al. (2003) verwendet, zum anderen zwei eigene Corpora erstellt, die speziell im Hinblick auf Fragen der Informativität konzipiert wurden. Die Aufgabe bei der Bewertung von Texteinheiten in Bezug auf ihre In-

⁸Da es sich bei EXCERPT um ein experimentelles System handelt, stand die Frage der einfachen und gewinnbringenden Nutzbarkeit (Bedienbarkeit) der angebotenen Parametrisierungsmöglichkeiten nicht im Zentrum des Interesses.

formativität folgte dabei einer Fragestellung von Hovy: "Ask experts to underline and extract the most interesting or informative fragments of the text. Measure recall and precision of the system's summary against the human's extract ..." (Hovy, 2004). Insgesamt wurden im eigenen Korpus 26 Texte verschiedener Textsorten von jeweils 13 Testpersonen hinsichtlich ihrer relevanten und informativen Sätze beurteilt: Von den 26 Texten wurden 10 für den generischen Summary-Modus (ohne Suchterme) auf informative Sätze hin bewertet, bei den anderen 16 Texten wurden ein bis drei Suchbegriffe vorgegeben, hinsichtlich deren die (i) relevantesten und (ii) die daraus informativsten Sätze ermittelt werden sollten citereischer:10. Zusammen mit den 13 Texten aus dem Zechner- und Hasler-Korpus standen somit 23 Texte im Summarizing- und 16 Texte im Within-Document-Retrieval-Modus zur Verfügung.⁹

Dabei wurde das Zechner-Korpus und partiell das erste eigene Korpus zusammen mit bewerteten Texten aus Vortests bei der Entwicklung und Kalibrierung des Systems verwendet; die anderen Corpora dienten als Testdatenmenge zur Überprüfung der Leistung von EXCERPT. Als einheitliches Leistungsmaß für die beiden Modi des Summarizings und Within-Document-Retrievals wurde das R-Precision-Maß verwendet (Baeza-Yates and Ribeiro-Neto, 1999; Manning et al., 2008), das den Anteil der vom System korrekt ermittelten Sätze mit der Gesamtmenge der von den Testpersonen als signifikant (relevant und informativ) beurteilten Sätze in Beziehung setzt. Ein Satz wurde dann als signifikant gewertet, wenn er von mehr als der Hälfte bzw. drei Viertel der Bewerter als relevant und/oder informativ beurteilt wurde ($7/13 = 54\%$ bzw. $10/13 = 77\%$).¹⁰ Im Within-Document-Retrieval-Modus ergibt sich eine Performance von etwa 0.75, d. h. drei von vier Sätzen wurden vom System korrekt ermittelt. Auf einen Vergleich mit

⁹In Anhang B findet sich ein Beispiel für einen bewerteten Text aus dem eigenen Korpus mit zwei Suchbegriffen samt den Wertungen auf Relevanz und Informativität.

¹⁰Beim Zechner-Korpus wurden effektiv ebenfalls 13 Testpersonen gefragt; beim Hasler-Korpus waren es für sieben Texte drei Personen, so dass hier lediglich die Zweidrittel-Mehrheit gebildet werden konnte. Das Konzept des Mehrheitsentscheids wurde gewählt, da das Maß der Übereinstimmung zwischen den Bewertern erwartbar niedrig war. Eine Mehrheitsentscheidung garantiert jedoch den besten Kompromiss bei der Auswahl von Texteinheiten, da hierdurch die meisten Nutzer zufriedengestellt werden können. Der Anspruch eines 'idealen' Summarys mit entsprechend hohen Übereinstimmungsraten bei Annotatoren ist m. E. von Grund auf verfehlt.

anderen Systemen musste mangels Verfügbarkeit verzichtet werden. Im Summarizing-Modus hingegen konnte zudem ein Vergleichstest mit drei teils kommerziellen Summarizern durchgeführt werden: dem Copernic, Intellexer- und SubjectSearch-Summarizer.¹¹ Dabei zeigte sich, dass EXCERPT mit einer durchschnittlichen Gesamt-Performance von ca. 0.55 am besten abschnitt, wobei die anderen Systeme im günstigsten Falle nicht über ca. 0.45 hinauskamen.

4 Fazit

Die Realisierung eines integrierten Summarizing- und Within-Document-Retrieval-Systems mit Fokus auf der Selektion informativer Textpassagen ist konzeptionell möglich und in ein reales Informationssystem umsetzbar. Die Performance des Systems übersteigt dabei die Leistung gängiger Summarizer, obgleich dies nicht der ursprüngliche Anspruch des Systems war. Als neuer Systemtyp zur inhaltlichen Arbeit mit Texten stellt das EXCERPT-System insoweit eine Innovation dar, als es eine Suchmaschine für Texte bietet, die zudem eine integrierte Summarizing-Funktionalität enthält. Die Erkenntnisse aus dem Bereich des automatischen (informativen) Summarizings werden dabei mit eigenen Untersuchungen zur Informativität von Texten kombiniert und auf Within-Document-Retrieval-Aufgaben übertragen. Der Implementierungsaufwand ist dabei nur unwesentlich höher als bei der Realisierung eines einfachen Recherche- oder Summarizing-Systems.

Der ideale Anwendungskontext für die Funktionalität des hier vorgestellten Informationssystems sind Dokumenten-Suchmaschinen, die dem Nutzer die sofortige Weitersuche und -auswahl von Textpassagen innerhalb eines Dokuments ermöglichen. Darüber hinaus bieten Webbrowser, PDF-Reader und Textverarbeitungen ein entsprechendes Umfeld, in dem Nutzer häufig nach spezifischer Information in Texten suchen müssen. Die Optimierung auf informative Textpassagen erspart dem Nutzer im Idealfall die Rezeption des gesamten Dokuments, da er inkrementell die wichtigsten Passagen angeboten bekommt. Die gleiche Technologie erlaubt auch nach entsprechender Anpassung

der Parametrisierung die Ausgabe inhaltlich repräsentativer oder indikativer (anstelle informativer) Textstellen, die charakteristisch für den Inhalt des gesamten Textes sind.

¹¹<http://www.copernic.com/en/products/summarizer>, <http://summarizer.intellexer.com>, <http://www.kryltech.com/summarizer.htm> (jeweils 20.4.2010). Eine ganze Reihe weiterer Summarizer wurde in Betracht gezogen, diese konnten jedoch aus verschiedenen Gründen nicht eingesetzt werden (z. B. verweigerten manche Systeme die Installation).

References

- E. Alfonseca and P. Rodríguez. 2003. Generating extracts with genetic algorithms. In *Proceedings of ECIR 2003*, pages 511–519.
- R. Baeza-Yates and B. Ribeiro-Neto. 1999. *Modern Information Retrieval*. Pearson, London u. a.
- M. Banko, V. Mittal, M. Kantrowitz, and J. Goldstein. 1999. Generating extraction-based summaries from hand-written summaries by aligning text-spans. In *Proceedings of the Pacific Association for Computational Linguistics PACLING*.
- R. Barzilay and M. Elhadad. 1999. Using lexical chains for text summarization. In I. Mani and M. T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 111–121. MIT Press, Cambridge und London.
- P. B. Baxendale. 1958. Machine-made index for technical literature – an experiment. *IBM Journal of Research and Development*, 2(4):354–361.
- H. Borko and C. L. Bernier. 1975. *Abstracting Concepts and Methods*. Academic Press, New York u. a.
- R.-A. de Beaugrande and W. Dressler. 1981. *Introduction to Text Linguistics*. Longman, London und New York.
- H. P. Edmundson. 1969. New methods in automatic extracting. *Journal of the American Society for Information Science*, 16(2):264–285.
- H. Elsen. 2004. *Neologismen. Formen und Funktionen neuer Wörter in verschiedenen Varietäten des Deutschen*. Narr, Tübingen.
- C. Fellbaum, editor. 1998. *WordNet – An Electronic Lexical Database*. MIT Press, Cambridge und London.
- R. Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.
- J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of SIGIR'99*, pages 121–128.
- U. Hahn and U. Reimer. 1986. Topic essentials. In *Proceedings of the 11th Conference on Computational Linguistics*, pages 497–503.
- D. J. Harper, I. Koychev, S. Yixing, and I. Pirie. 2004. Within-document-retrieval: A user-centred evaluation of relevance profiling. *Information Retrieval*, 7:265–290.
- L. Hasler, C. Orasan, and R. Mitkov. 2003. Building better corpora for summarisation. In *Proceedings of Corpus Linguistics*, pages 309–319.
- E. Hovy. 2004. Text summarization. In R. Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, pages 583–598. University Press, Oxford.
- C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. University Press, Cambridge.
- V. Mittal, M. Kantrowitz, J. Goldstein, and J. Carbonell. 1999. Selecting text spans for document summaries: Heuristics and metrics. In *Proceedings of AAAI-99*, pages 467–473.
- C. Paice. 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26(1):171–186.
- D. Radev. 2009. Bibliography summarization papers.
- J. Reischer. 2006. *Zeichen, Information, Kommunikation. Analyse und Synthese des Zeichen- und Informationsbegriffs*. <http://epub.uni-regensburg.de/10483/> (20.4.2010).
- J. Reischer. 2009. Excerpt – a within-document retrieval system using summarization techniques. In R. Kuhlen, editor, *Information Droege, Ware oder Commons? Wertschöpfungs- und Transformationsprozesse auf den Informationsmärkten*, pages 63–75. Proceedings des 11. Internationalen Symposiums für Informationswissenschaft (ISI 2009). Boizenburg: Verlag Werner Hülsbusch.
- J. Reischer. 2010. *Retrieval und Ranking informativer Textpassagen. Eine theoretische und praktische Integration von informativem Summarizing und Within-Document-Retrieval*. Habilitationsschrift, Universität Regensburg.
- G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.
- W. G. Stock. 2007. *Information Retrieval*. Oldenbourg, München und Wien.
- E. von Weizsäcker. 1974. Erstmaligkeit und bestätigung als komponenten der pragmatischen information. In C. F. von Weizsäcker, editor, *Die Einheit der Natur*, pages 82–113. dtv, München.
- K. Zechner. 1995. Automatic text abstracting by selecting relevant passages. M.Sc. dissertation, Edinburgh.

A. Beispiel-Suche

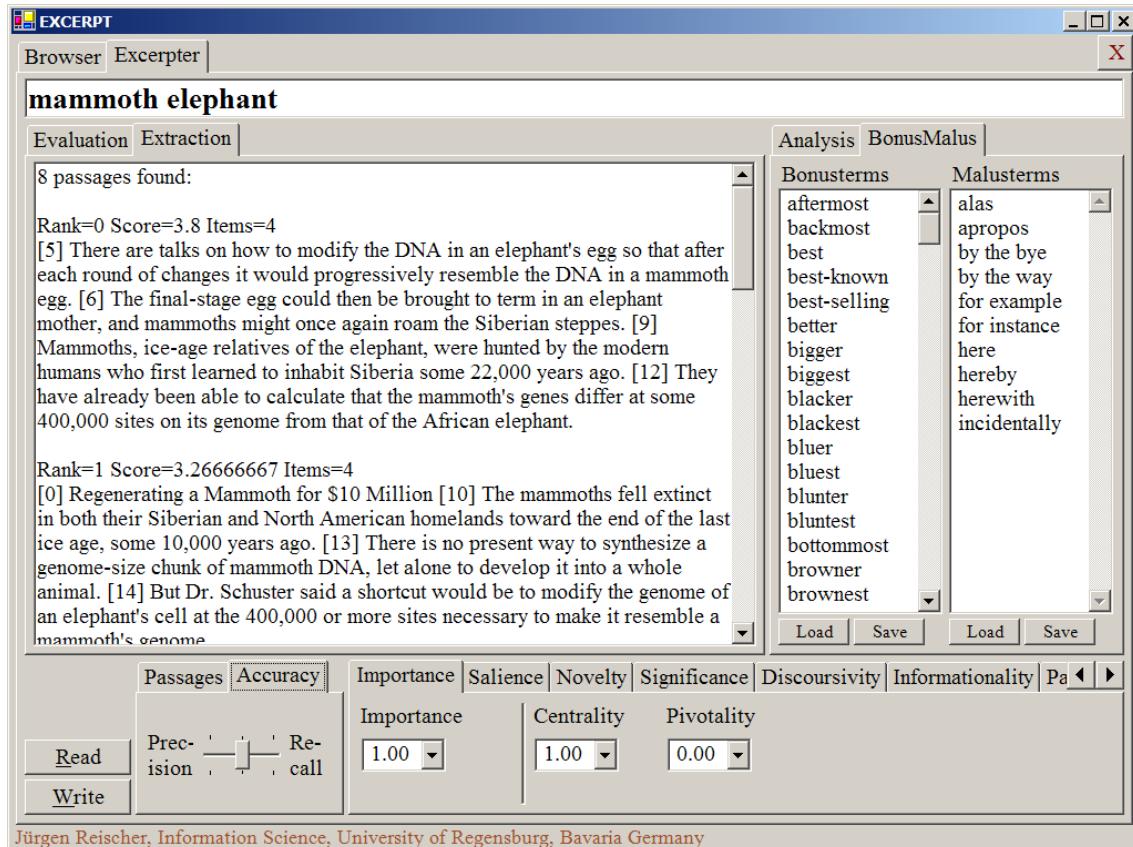


Abbildung 1: Beispiel-Suchanfrage an den Text aus Anhang B und die durch das EXCERPT-System ermittelte Ausgabe in Form einer gerankten Liste von Summarys

B. Beispiel-Extrakt

Das folgende Extrakt entstammt einem Text aus dem Evaluationskorpus zur Informativität von Texten (Reischer, 2010). In Klammern stehen die Satzindexe derjenigen Sätze, die von den Testpersonen als relevant und informativ bezüglich der Suchbegriffe "mammoth" und "elephant" ermittelt wurden (die Überschrift ist Satz [0]). Die genauen Anzahlen der Wertungen für alle Sätze finden sich nachstehend.

[5] There are talks on how to modify the DNA in an elephant's egg so that after each round of changes it would progressively resemble the DNA in a mammoth egg. [6] The final-stage egg could then be brought to term in an elephant mother, and mammoths might once again roam the Siberian steppes. [9] Mammoths, ice-age relatives of the elephant, were hunted by the modern humans who first learned to inhabit Siberia some 22,000 years ago. [12] They have already been able to calculate that the mammoth's genes differ at some 400,000 sites on its genome from that of the African elephant. [14] But Dr. Schuster said a shortcut would be to modify the genome of an elephant's cell at the 400,000 or more sites necessary to make it resemble a mammoth's genome.

Anzahl Wertungen durch die Testpersonen je Satz:

[0]: 3, [1]: 6, [2]: 3, [3]: 0, [4]: 3, [5]: 13, [6]: 12, [7]: 0, [8]: 3, [9]: 10, [10]: 3, [11]: 4, [12]: 10, [13]: 3, [14]: 9, [15]: 6, [16]: 1, [17]: 4, [18]: 2, [19]: 2, [20]: 2, [21]: 1, [22]: 1, [23]: 1, [24]: 1, [25]: 0, [26]: 1, [27]: 0, [28]: 0, [29]: 1.

Sätze, die von mehr als der Hälfte der Bewerter selektiert wurden, sind unterstrichen; Sätze, die entsprechend von drei Vierteln ausgewählt wurden, sind unterstrichen.

Der Volltext findet sich unter www.nytimes.com/2008/11/20/science/20mammoth.html (19.11.2008). Die vollständige Satzzerlegung kann beim Autor nachgefragt werden.

Using tf-idf-related Measures for Determining the Anaphoricity of Noun Phrases

Julia Ritz

Collaborative Research Centre 632 “Information Structure”

Universität Potsdam

Potsdam, Germany

jritz@uni-potsdam.de

Abstract

The study presented here tests the suitability of measures based on term frequency (tf) and inverse document frequency (idf) for distinguishing anaphoric (i.e. discourse-given) from non-anaphoric noun phrases in English newswire using a classifier. Results show these features can make a significant contribution without relying on external, language-specific resources like hand-crafted rules or lexical semantic networks.

1 Motivation

Tf-idf (term frequency times inverse document frequency) is a shallow semantic measure of the information content of a linguistic entity (e.g. a word). It is commonly used in information retrieval, document management, and text summarization (Salton and Buckley, 1988; Neto et al., 2000). The goal of this study is to investigate its usefulness in determining the anaphoricity of noun phrases (NPs) in discourse. This is, to the best of my knowledge, a new field of application.

By anaphoricity, I mean an NP’s property of being either anaphoric, i.e. referring to a real-world entity mentioned previously (in the left context), or else non-anaphoric (a more detailed definition is given in Section 3.1). Anaphoricity classification can serve as a first step in anaphora resolution: the ruling out of non-anaphoric NPs helps limiting the search space and can thus improve a system’s quality and performance (Harabagiu et al., 2001; Ng and Cardie, 2002; Elsner and Charniak, 2007; Uryupina, 2009; Rahman and Ng, 2009; Ng, 2010; Poesio et al., 2004; Kabadjov, 2007, the latter two refer to definite descriptions). One of the main challenges identified in previous work on anaphoricity classification and anaphora resolu-

tion is the inaccuracy of methods that try to match potential anaphor-antecedent pairs. Various strategies have been proposed, including matching the complete NP strings, their respective heads, their first or last tokens, etc. (Bergler, 1997; Soon et al., 2001; Uryupina, 2003; Uryupina, 2009; Nissim, 2006). Relying on token boundaries, these methods typically have difficulties with compounding (*Miami-based* in Example 1), derivation (e.g. event anaphors like (2)), and variations of names (see Example 3).¹

- (1) Four former Cordis Corp. officials were acquitted of federal charges related to the Miami-based company’s sale of pacemakers, including conspiracy to hide pacemaker defects. Jurors in U.S. District Court in Miami cleared Harold Hershenson.
- (2) Weatherford International Inc. said it canceled plans for a preferred-stock swap. Weatherford said market conditions led to the cancellation of the planned exchange.
- (3) A Coors spokesman said the company doesn’t believe the move will reduce the influence of Jeffrey Coors₁, Peter Coors₂ or Joseph Coors Jr.₃, who run the company’s three operating units. “Pete₂ and Jeff₁ and Joe Jr.₃ have taken over the reins and are doing most of the work.”

In contrast to this, I employ a matching based on term² frequency weighted by inverse document frequency. This method captures similarity

¹All examples and their annotations taken from the OntoNotes corpus (Hovy et al., 2006), slightly shortened.

²By term, I mean an ngram of characters. In my experiments, term length was set to 4 characters as suggested by, e.g., Taboada et al. (2009).

between strings, taking less account of exact token boundaries as it matches terms, i.e. parts of strings, rather than tokens. At the same time, it considers the informational content (or specificity) of a string in context to the effect that it typically ranks a) the contribution of function words, affixes, etc. lower than that of content words, and b) the contribution of highly specific terms (occurring in few documents) higher than that of common terms (occurring in a large portion of the documents).

This paper represents an evaluation of this strategy in comparison to other matching strategies.

It is structured as follows: In Section 2, I will introduce the features used for anaphoricity classification. Section 3 reports on the classification experiments and their results. The proposed methods will be discussed in Section 4, and conclusions are drawn in Section 5.

2 Tf-idf adapted for anaphoricity classification

For each NP to be classified with respect to anaphoricity, I calculate a set of features based on tf-idf. Commonly, tf-idf of a term t in a document d is calculated as in (4), with D the document collection, D_t the set of documents containing t , and $tf_{t,d}$ the number of times t is contained in document d (normalized to the total number of terms in d).

$$tfidf_{t,d} = tf_{t,d} * \log\left(\frac{|D|}{|D_t|}\right) \quad (4)$$

For the special purpose of anaphoricity classification, I partition the text right before the mention of the NP currently under investigation: Let X be a text (i.e. one document) consisting of the characters x_1, \dots, x_n , and NP be a (noun) phrase starting at character position $k + 1$. I then define tf_{t,d_k} as the relative frequency of t in d_k (i.e. document d up to position k). Subsequently, the increase of tf after k is calculated as follows:

$$tf_{t,\overline{d_k}} = tf_{t,d} - tf_{t,d_k} \quad (5)$$

To adapt to NPs as units of interest, I calculate the sum (as well as the means) of each measure ($tf_{t,d}$, tf_{t,d_k} , $tf_{t,\overline{d_k}}$, $tfidf_{t,d}$, $tfidf_{t,d_k}$, $tfidf_{t,\overline{d_k}}$, and idf) across the NP. Let NP_s^e be an NP starting at character position s and ending at character position e . Then, the sum across this NP is defined

as

$$s_{tfidf_{NP_s^e, d_s}} = \sum_{i=s}^{e-l+1} tfidf_{t_i, d_i} \quad (6)$$

with l the predefined term length (here set to 4). The means is calculated analogously. Calculating the means of tf-idf across a phrase (here: an NP) is a common method to obtain sentence relevances in text summarization, cf. Bieler and Dipper (2008).

It should be noted that features based on $tf_{t,d}$ also take into account the right context of an anaphor candidate. The right context usually is not considered in anaphora resolution and related tasks; however, it has been found helpful in manual annotation experiments (Stührenberg et al., 2007).

3 Classification Experiments

3.1 Corpus and Annotation Scheme

As a test corpus, I used the Wall Street Journal section of OntoNotes (Hovy et al., 2006; Weischedel et al., 2007).

The OntoNotes task of coreference annotation is to “connect [...] coreferring instances of specific referring expressions, meaning primarily NPs that introduce or access a discourse entity” (Hovy et al., 2006). The scheme distinguishes two relations: APPOS for the linking attributive/appositive NPs, and IDENT for other coreference links. The definition of coreference includes coreference between individual objects (see Examples 1 and 3 in Section 1), activities, situations, states, or sets of any of these, as well as event anaphora (Example 2), without further distinguishing different types of coreference. It excludes (i.e. leaves unannotated) generic, underspecified, and abstract NPs (in particular, references to the same type or concept, see dashed underlined NPs in Example 7), as well as bridging relations.

- (7) Medical scientists are starting to uncover a handful of genes which, if damaged, unleash the chaotic growth of cells that characterizes cancer. Scientists say the discovery of these genes in recent months is painting a new and startling picture of how cancer develops.

As to annotation quality, Hovy et al. (2006) report “average agreement scores between each annotator and the adjudicated results [of] 91.8%”.

I employed the syntax annotation for extracting noun phrases, and the coreference annotation ('IDENT relation') for inducing the class of an NP: anaphoric (AN) if it has an antecedent (i.e. an expression to its left referring to the same ID), or else non-anaphoric (NON). Any unannotated NP is considered non-anaphoric here.

3.2 Preprocessing and Settings

The documents were randomly split into a training and an evaluation set, controlled with respect to the number of NPs contained in each document.

Table 1 shows the numbers of AN and NON NPs in the two sets, including the same information on a subset of the data consisting of only those NPs containing proper names (NNPs), for the purpose of a more specific evaluation.

Set	NPs	AN	NON
Train	103,245	13,699 (13.27%)	89,546 (86.73%)
Test	13,414	1,828 (13.63%)	11,586 (86.37%)
Train _{NNP}	18,324	4,873 (26.59%)	13,451 (73.41%)
Test _{NNP}	2,332	649 (27.83%)	1,683 (72.17%)

Table 1: Overview of Anaphoricity Distribution in Training and Test Set, respectively.

The features serving as a comparison to tf-idf-based features are (i) exact match (abbreviated as 'exact'), a numeric feature counting the number of times the current NP has been previously mentioned in exactly the same way (string identity), and (ii) matching head (abbreviated 'head'), another numeric feature counting the number of times the lemma of the current NP's head occurs in the lemmatized context (disregarding their parts of speech). The lemmatization was performed using TreeTagger (Schmid, 1994); here, the head is the rightmost token directly dominated by the NP node. Features describing the linguistic form (abbreviated 'ling. form') are all nominal features extracted from the parts of speech and syntactic annotation (or, in the case of pronoun and determiner form, the actual surface form): these features include (i) the NP's grammatical function (subject, adverbial NP, etc.), (ii) its surface form (pronoun, definite determiner, indefinite determiner, etc.),

(iii) whether it contains a name (NNP), and (iv) –in the case of pronouns– morphological features (person, number, gender, reflexivity).

3.3 Classification Experiments and Results

For training different classifiers and evaluating the results, I used the data mining software WEKA (Witten and Frank, 2005). C4.5 decision trees were trained on the training set and the resulting models were applied to the test set. The baseline was obtained by always assigning the majority class (NON), which results in an accuracy of 86.37%.

Classification results are shown in Tables 2 and 3³, showing the features' performance in combination as well as in contrast to others. Table 2 documents an additive evaluation. Results in this table show a significant improvement in accuracy with the addition of each feature. Tf-idf-related features especially improve the recall of finding anaphoric NPs (more than twice as compared to feature sets (1) and (2), and nearly twice the F measure). Table 3 contains the results of a contrastive study of different features. The left part of the table contains results based on all NPs; the right part contains results based on a specific subgroup of NPs from the training and test set, respectively, namely those containing names (NNPs). Results show that generally (left part of table), tf-idf-related features alone do not perform any different from the baseline. On the NNP subset (right part of table), however, they perform substantially better than the baseline. From the leave-one-out experiments, we see that in general, the exclusion of any of the features has a similar effect (differences between these classifiers are not significant). On the NNP subset, the exclusion of the feature 'head' seems to have the greatest impact (mirroring the common strategy of referring to a mentioned person using their last name).

In conclusion, all features presented have a certain overlap, but nevertheless perform best when combined.

4 Discussion

4.1 Related Work

The task of anaphoricity prediction has been approached in previous studies: Uryupina (2003) classifies noun phrases as ±discourse_new at an

³Abbreviations: Acc(uracy), Precision), R(ecall), F(measure), AN(aphoric), NON(anaphoric).

set	feature(s) used	acc	P	R	F	value
(0)	majority class	86.37%				
(1)	exact	87.38% ⁰	86.40%	100.00%	92.70%	NON
(2)	(1) + head	88.65% ^{0,1}	89.00%	97.50%	93.00%	NON
(3)	(2) + tf-idf feats	90.44% ^{0,1,2}	88.90%	99.30%	93.80%	NON
(4)	(3) + ling. form	93.20% ^{0,1,2,3}	92.80%	96.40%	94.60%	NON
			95.30%	97.00%	96.10%	NON

Table 2: Classification results (additive evaluation).³

⁰ number n in superscript: significant improvement in accuracy compared to feature set number n (McNemar test, level of significance: $p < .05$)

feats	acc (all NPs)		acc (NNPs only)	
	single	leave-out	single	leave-out
exact	87.38% ^{+, -}	88.88% ^{+, -}	83.45% ^{+, -}	83.36% ^{+, -}
head	87.92% ^{+, -}	88.52% ^{+, -}	84.43% ^{+, -}	82.85% ^{+, -}
tf-idf	86.22% ⁻	88.65% ^{+, -}	77.06% ^{+, -}	84.73% ^{+, -}
(0)	86.37%		72.17%	
(3)		90.44%		86.11%

Table 3: Classification results (contrastive evaluation).³

Comparison to baseline and feature set (3) in superscript.

⁺ significant improvement over baseline (0)

⁻ significant decline compared to (3)

(McNemar test, level of significance: $p < .05$)

accuracy of 81.12% on 3,710 NPs from the MUC-7 corpus (5-fold cross-validation). Nissim (2006) reports an accuracy of 93.1% (distinguishing old vs. mediated/new NPs) on an evaluation set of 12,624 NPs from the Switchboard corpus, a corpus of transcribed dialogues (the training set consisting of 40,865 NPs). Elsner and Charniak (2007) present a discourse-new classifier with 82.63% accuracy on a test set of 2,505 NPs from the MUC-7 data. Kabadzov (2007) reports on discourse-new classifiers for definite descriptions based on 1,848 NPs from the GNOME corpus (museum and pharmaceutical domain) and 4,842 NPs from the VPC corpus (Wall Street Journal). He experiments with different classification algorithms; the best results are 86.9% accuracy (10-fold cross-validation).

Yet, it should be noted that none of the results are directly comparable due to differences in the data as well as the annotation schemes.

Finally, as to the related field of coreference resolution, Yang et al. (2004) have used tf-idf (with a term being a token) as a factor for weighting tokens in the comparison of mention-pairs (i.e. an anaphor and its potential antecedent). Experimental results with different string distance metrics, however, do not indicate an improvement of overall performance in the F-measure when using tf-idf as a weighting factor in these metrics as compared to when using a binary weighting.

4.2 Complexity

An algorithm for calculating the measures presented in Section 2 is given in Figure 1.⁴ It performs in linear time ($O(n)$ with n the number of characters in the corpus) given that hashes have

⁴Hashes are denoted by a capital letter followed by the key(s) in brackets, e.g. $X(k_1, k_2)$.

linear access time.⁵ The number of different terms is $l|\Sigma|$ (with term length l , and alphabet Σ), and can be significantly limited by converting characters to lowercase and mapping special characters to a specific character, e.g. underscore ('_'). The corresponding code could be inserted after the lines indicated by '*' in Fig. 1.

```

#1st pass
l:=4; #initialize term length l
D:=0; #initialize file counter D
for each Document  $d_i$  in the corpus
    #count document
    D++;
    p:=1; #initialize character position p
    while  $p + l$  in  $d_i$ 
        #sequentially cut into terms  $t$  of length  $l$ 
        t:=substring( $d_i$ ,  $p$ ,  $l$ );
        #*insert string normalization (optional)*
        #initialize count array where necessary
        C( $t$ ,  $d_i$ ):=0 unless defined;
        #save number of previous mentions
        #(i.e. annotate  $t$  with  $C(t, d_i)$ )
        A( $t$ ,  $d_i$ ,  $p$ ):=C( $t$ ,  $d_i$ );
        #count current mention
        C( $t$ ,  $d_i$ )++;
        #count documents containing  $t$ 
        #(only on first mention of  $t$ )
        E( $t$ )++; if ( $C(t, d_i) = 1$ );
        p++;
    end; #end while
end; #end for each;
#2nd pass
for each Document  $d_i$  in the corpus
    for each noun phrase  $NP_s^e$  in  $d_i$ 
        sum:=0; #initialize sum
        #from NP's starting position...
        p:=s;
        #... to start of last term
        while  $p <= e - l + 1$ 
            t:=substring( $d_i$ ,  $p$ ,  $l$ );
            #*insert string normalization (optional)*
            #get annotation of  $t$  at  $p$ ,
            #calculate tf-idf from it
            #and add it to the current sum
            sum+=(get( $t$ ,  $d_i$ ,  $p$ )/ $p$ )*log( $D/E(t)$ );
            #calculate sum of other measures
            ...
        end; #end while
        #average by the number of terms in  $NP_s^e$ 
        a:=sum/( $e - s - l + 2$ );
        #annotate sum and means to  $NP_s^e$ 
        S( $d_i$ ,  $s$ ,  $e$ ):=sum;
        M( $d_i$ ,  $s$ ,  $e$ ):=a;
    end; #end for each
end; #end for each

```

Figure 1: Algorithm for calculating tf-idf-based measures⁴

⁵The complexity of the second pass also depends on the maximal depth c of NP embedding in the corpus ($O(c * n)$). In practice, however, c will be of little consequence, as we ignore material outside NPs in this pass.

4.3 Error Analysis

The results from a classification using all features were re-annotated to the corpus and imported into the corpus interface ANNIS (Chiarcos et al., 2009; Zeldes et al., 2009), where they can be queried and viewed in context, including links to antecedents (where antecedents exist). The total number of classification errors is 912 (corresponding to an error rate of 6.80%). 559 (4.17% of all NPs) were misclassified as NON and 353 NPs (2.63% of all NPs) as AN. An error analysis of a 10% sample is presented in Table 4⁶. The first column contains the absolute number of errors in the sample, the second contains a confidence interval estimating the minimal and maximal benefit that a solution for this error class would achieve: E.g. there are 16 anaphors in the ‘NON’ sample that have a semantic relation to the context, amounting to an estimated maximum of 40.41% in the whole test set. In the best-case scenario (with a perfectly-performing semantic component at hand), the error rate among NON NPs (4.17%) could be reduced by 40.41% to 2.48%. All other parameters unchanged, the overall error rate would be 5.11% (2.48% in the category NON + 2.63% in AN). This means, theoretically, the classifier’s accuracy could be improved to 94.89%.⁷ Among the first error group, NPs misclassified as NON, the largest subgroup is one where at least one of the tokens of anaphora and antecedent are identical. The classifier typically failed because these tokens a) make up a rather small proportion of the NP (see second Example ‘sites’ in first group (named ‘similarity’), Table 4), or b) are hardly specific (e.g. the heads in *the market, this year*). This subgroup also contains many deictic references, i.e. references to temporal, spacial or personal aspects of the utterance situation, which can be resolved independently from the previous text.

Among the second group of errors, NPs misclassified as AN, there is a considerable proportion of entities that are – if not anaphoric – at least related to the context (via bridging or a more vague reference to the situation described in the context). A second subgroup is deictic reference again, which is resolvable via the situational context of the utterance. These two subgroups to-

⁶Abbreviations: abs - absolute number, ci - confidence interval ($\alpha < 0.05$).

⁷This estimation disregards the fact that semantic similarity may well hold between non-coreferent entities.

misclassified as non

abs	ci	
20	[23.16%;48.24%]	relation anaphora - antecedent (examples in brackets) similarity (though marginal or unspecific to the respective text), e.g. identical heads (<i>the coast</i> - <i>the eastern coast of Texas</i> ; <i>four potential sites for its operations</i> - <i>sites in Austin, Texas; Colorado Springs, Colo.; Middletown, N.Y.; and Phoenix, Ariz.</i>), 9 of these [6.45%;25.69%] are deictic references (references to location, time, or interlocutors of the respective utterance situations, like <i>the coming year</i> - <i>next year</i> ; <i>the country</i> - <i>this country</i>)
7	[3.84%;21.16%]	variations of proper names (<i>Frank W. Terrizzi</i> - <i>Mr. Terrizzi</i> ; <i>U.S. Memories Inc.</i> - <i>U.S. Memories</i>)
6	[2.61%;18.79%]	annotation errors, 5 [1.43%;16.37%] of them possessive relations (<i>Times magazine_i</i> - <i>the magazine_{i(correct)}</i> 's executive editor _{i(incorrect)})
6	[2.61%;18.79%]	head noun more specific than antecedent (<i>Friday's action</i> - <i>Friday's big drop</i> ; <i>My language</i> - <i>my English</i>)
6	[2.61%;18.79%]	head noun more general (<i>black skimmers</i> - <i>the shimmering fish, knocking share prices down</i> - <i>the action</i>)
4	[0.38%;13.82%]	synonymous anaphor (<i>U.K. trade figures</i> - <i>The trade and CBI reports</i>)
2	[0%;8.88%]	abbreviation (<i>the recently formed Resolution Trust Corp.</i> - <i>the RTC</i>)
2	[0%;8.88%]	antecedent is cataphoric (<i>it</i>)
2	[0%;8.88%]	variations of dates (<i>Oct. 19, 1987</i> , - <i>October '87</i>)
1	[0%;5.26%]	switch to or from direct speech (" <i>My language ...</i> " - <i>the [...] guide</i>)
56		(first 10% of 559)
(16	[16.73%;40.41%]	of these are semantically related: synonyms, hypernyms, hyponyms)

misclassified as an

abs	ci	NPs mistaken for anaphors
7	[6.75%;33.25%]	annotation errors (non-maximal projections, including 3 embedded genitive anaphors (<i>Friday_i</i> - <i>Friday_{i(missing)}</i> 's fall) and 2 NPs where the antecedent was mentioned in an apposition)
6	[5.30%;28.98%]	generic (<i>the dollar has been strong</i>)
4	[3.10%;19.76%]	bridging relation to referent(s) in the context (member of the same set/aggregation), e.g. <i>an \$89.6 million Navy contract</i> - <i>a \$75.5 million Navy contract</i>
4	[3.10%;19.76%]	vaguely anaphoric to the general situation described in the context (<i>If the dollar stays weak, [...] that will add ...; the economy's slower growth this year [...] So it's a very mixed bag.</i> ; ...)
4	[3.10%;19.76%]	deictic ('we' in direct speech, 'earlier this year')
3	[1.24%;15.90%]	expletive (' <i>it seems to me</i> ' etc.)
1	[0%;7.23%]	cataphoric (<i>As she calculates it, the average stock...</i>)
6	[5.30%;28.98%]	other similarities (such <i>sentiment</i> - <i>The market sentiment</i> , <i>Black Monday</i> - <i>Monday</i>)
35		(first 10% of 353)
(10	[13.60%; 43.54%]	of 35 are pronouns)
(8	[8.94%; 36.77%]	of 35 occur in direct speech)

Table 4: Error analysis of a sample of the classification errors.⁷

gether have been categorized in the literature as *discourse-new hearer-old* (Prince, 1981; Prince, 1992), *mediated* (Nissim et al., 2004) or *accessible* (Dipper et al., 2007; Riester et al., 2010). Publicly available resources taking this category into account remain a desideratum.

In either group, there is a small proportion of annotation errors (presumably an embedding issue of the XML linearization of the original annotations).

5 Conclusion and Outlook

Anaphoricity classification significantly profits from tf-idf-related features.

In contrast to other features (head match, etc.), they can be obtained without preprocessing the data (lemmatising, head identification). Results on name anaphors suggest that a classification is feasible even in the absence of the other features. This makes tf-idf-related features promising for scarcely-resourced languages;

a cross-linguistic evaluation remains to be done.

Regarding further improvements to the measures, also calculating the maximum of tf-idf across an NP might be helpful.

Finally, a finer-grained annotation to train on, with one or more categories in between anaphoric and non-anaphoric (e.g. the category accessible, see previous section for references), could lead to a theoretically better-motivated classification model of anaphoricity.

Acknowledgements

I would like to thank everyone who helped write this paper by reviewing it, in particular by pointing out related literature.

References

- Sabine Bergler. 1997. Towards Reliable Partial Anaphora Resolution. In *ANARESOLUTION '97: Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 62–66, Morristown, NJ, USA. Association for Computational Linguistics.
- Heike Bieler and Stefanie Dipper. 2008. Measures for Term and Sentence Relevances: an Evaluation for German. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapia, editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Christian Chiarcos, Stefanie Dipper, Michael Götze, Ulf Leser, Anke Lüdeling, Julia Ritz, and Manfred Stede. 2009. A Flexible Framework for Integrating Annotations from Different Tools and Tagsets. *TAL (Traitement automatique des langues)*, 49(2):271–293.
- S. Dipper, M. Götze, and S. (Eds.) Skopeteas. 2007. Information Structure in Cross-Linguistic Corpora: Annotation Guidelines for Phonology, Morphology, Syntax, Semantics, and Information Structure. Technical report, University of Potsdam.
- Micha Elsner and Eugene Charniak. 2007. A generative discourse-new model for text coherence. Technical Report CS-07-04, Brown University, Providence, RI, USA.
- Sanda M. Harabagiu, Răzvan C. Bunescu, and Steven J. Maiorano. 2001. Text and Knowledge Mining for Coreference Resolution. In *Proceedings of NAACL-01*, pages 55–62, Pittsburgh, PA.
- Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.
- Mijail Alexandrov Kabadzov. 2007. *A Comprehensive Evaluation of Anaphora Resolution and Discourse-new Classification*. Ph.D. thesis, University of Essex, U.K.
- Joel Larocca Neto, Alexandre D. Santos, Celso A. A. Kaestner, and Alex A. Freitas. 2000. Document Clustering and Text Summarization. In *4th International Conference on Practical Applications of Knowledge Discovery and Data Mining*, pages 41–55, London. The Practical Application Company.
- Vincent Ng and Claire Cardie. 2002. Identifying Anaphoric and Non-Anaphoric Noun Phrases to Improve Coreference Resolution. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*, pages 730–736. ACL.
- Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden, July. Association for Computational Linguistics.
- Malvina Nissim, Shipra Dingare, Jean Carletta, and Mark Steedman. 2004. An Annotation Scheme for Information Status in Dialogue. In *Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004)*, pages 1023–1026, Lisbon, Portugal.
- Malvina Nissim. 2006. Learning Information Status of Discourse Entities. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language*

- Processing (EMNLP 2006)*, pages 94–102, Sydney, Australia.
- Massimo Poesio, Olga Uryupina, Renata Vieira, Miguel Alex, Rodrigo Goulart, and Computao Aplicada (brazil). 2004. Discourse-new detectors for definite description resolution: a survey and preliminary proposal. In *In Proceedings of the Reference Resolution Workshop at ACL04*.
- Ellen F. Prince. 1981. Toward a taxonomy of given-new information. In Peter Cole, editor, *Radical Pragmatics*, pages 223–255. Academic Press, New York.
- Ellen F. Prince. 1992. The ZPG Letter: Subjects, Definiteness, and Information-Status. In William C. Mann and Sandra A. Thompson, editors, *Discourse Description. Diverse linguistic analyses of a fundraising text*, pages 295–325. Benjamins, Amsterdam.
- Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of EMNLP*, pages 968–977.
- Arndt Riester, David Lorenz, and Nina Seemann. 2010. A recursive annotation scheme for referential information status. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).
- Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.
- Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK. Extended version available at <http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf>.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.
- Maik Stührenberg, Daniela Goecke, Nils Diewald, Irene Cramer, and Alexander Mehler. 2007. Web-based Annotation of Anaphoric Relations and Lexical Chains. In *Proceedings of The Linguistic Annotation Workshop (LAW)*, pages 140–147, Prague, Czech Republic. Association for Computational Linguistics.
- Maite Taboada, Julian Brooke, and Manfred Stede. 2009. Genre-Based Paragraph Classification for Sentiment Analysis. In *Proceedings of the 10th SIGdial Workshop on Discourse and Dialog*, London, UK.
- Olga Uryupina. 2003. High-precision Identification of Discourse New and Unique Noun Phrases. In *Proceedings of the ACL Student Workshop*, Sapporo, Japan.
- Olga Uryupina. 2009. Detecting Anaphoricity and Antecedenthood for Coreference Resolution. In *Procesamiento del lenguaje natural*, volume 42, pages 113–120. Sociedad Española para el Procesamiento del Lenguaje Natural.
- Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Linnea Micciulla, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Olga Babko-Malaya, Eduard Hovy, Robert Belvin, and Ann Houston. 2007. OntoNotes Release 1.0. Technical report, Linguistic Data Consortium, Philadelphia.
- Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2004. Improving noun phrase coreference resolution by matching strings. In *In Proceedings of 1st International Joint Conference of Natural Language Processing*, pages 326–333.
- Amir Zeldes, Julia Ritz, Anke Lüdeling, and Christian Chiarcos. 2009. ANNIS: A Search Tool for Multi-Layer Annotated Corpora. In *Proceedings of Corpus Linguistics 2009*, Liverpool, UK.

Natural Language Processing for the Swiss German dialect area

Yves Scherrer

LATL

Université de Genève
Genève, Switzerland

yves.scherrer@unige.ch

Owen Rambow

CCLS

Columbia University
New York, USA

rambow@ccls.columbia.edu

Abstract

This paper discusses work on data collection for Swiss German dialects taking into account the continuous nature of the dialect landscape, and proposes to integrate these data into natural language processing models. We present knowledge-based models for machine translation into any Swiss German dialect, for dialect identification, and for multi-dialectal parsing. In a dialect continuum, rules cannot be applied uniformly, but have restricted validity in well-defined geographic areas. Therefore, the rules are parametrized with probability maps extracted from dialectological atlases.

1 Introduction

Most work in natural language processing is geared towards written, standardized language varieties. This focus is generally justified on practical grounds of data availability and socio-economical relevance, but does not always reflect the linguistic reality. In this paper, we propose to include continuous linguistic variation in existing natural language processing (NLP) models, as it is encountered in various dialect landscapes.

Besides continuous variation on the geographical axis, dialects represent some interesting challenges for NLP. As mostly spoken language varieties, few data are available in written form, and those which exist do not follow binding spelling rules. Moreover, dialect use is often restricted to certain social contexts or modalities (diglossia), reducing further the availability of resources.

In contrast, two facts facilitate the development of NLP models for dialects. First, dialects are generally in a historic and etymological relationship with a standardized language variety for which linguistic resources are more readily accessible. Sec-

ond, many dialects have been studied systematically by dialectologists, and these results can be exploited in a computational setting. The work presented here is applied to Swiss German dialects; this dialect area is well documented by dialectological research and is among the most vital ones in Europe in terms of social acceptance and media exposure.

This paper introduces ongoing work on a rule-based system that accounts for the differences between Standard German and the Swiss German dialects, using rules that are aware of their geographical application area. The system proposed here transforms morphologically annotated Standard German words into Swiss German words depending on the dialect area. The obvious use case for these components is (word-by-word) machine translation, which will be described in section 5.1. We also present two other applications that indirectly rely on these components, dialect identification (Section 5.2) and dialect parsing (Section 5.3).

We will start by presenting some related work (Section 2) and by giving an overview of the particularities of Swiss German dialects (Section 3). In Section 4, we present original work on data collection and show how probabilistic maps can be extracted from existing dialectological research and incorporated in the rule base. Then, the applications introduced above will be presented, and the paper will conclude with the discussion of some preliminary results.

2 Related work

Several research projects have dealt with dialect machine translation. The most similar work is the thesis by Forst (2002) on machine translation from Standard German to the Zurich Swiss German di-

alect within the LFG framework. Delmonte et al. (2009) adapt recent statistical machine translation tools to translate between English and the Italian Veneto dialect, using Standard Italian as a pivot language. In contrast, we are interested in handling a continuum of dialects.

Translation between dialectal variants can be viewed as a case of translation between closely related languages. In this domain, one may cite works on different Slavic languages (Hajic et al., 2003) and on the Romance languages of Spain (Corbí-Bellot et al., 2005).

Dialect parsing models have also been developed in the last years. Chiang et al. (2006) build a synchronous grammar for Modern Standard Arabic and the Levantine Arabic dialect. Their approach is essentially corpus-driven on the Standard Arabic side, but includes manual adaptations on the dialect side. Vaillant (2008) presents a factorized model that covers a group of French-based Creole languages of the West-Atlantic area. His model relies on hand-crafted rules within the TAG framework and uses a numeric parameter to specify a particular dialect.

With the exception of Vaillant (2008), the cited papers only deal with one aspect of dialect NLP, namely the fact that dialects are similar to a related standardized language. They do not address the issue of interdialectal variation. Vaillant's factorized model does deal with several related dialects, but conceives the different dialects as discrete entities which can be clearly distinguished. While this view is probably justified for Caribbean creoles spoken on different islands, we argue that it cannot be maintained for dialect areas lacking major topographical and political borders, such as German-speaking Switzerland.

One important part of our work deals with bilingual lexicon induction. For closely related languages or dialects, cognate words with high phonetic (or graphemic) similarity play a crucial role. Such methods have been presented in various contexts, e.g. by Mann and Yarowsky (2001), Koehn and Knight (2002), or Kondrak and Sherif (2006). Scherrer (2007) uses similarity models based on learned and hand-crafted rules to induce Standard German – Bern Swiss German word pairs.

Dialect identification has usually been studied from a speech processing point of view. Biadsy et al. (2009) classify speech material from four Arabic dialects plus Modern Standard Arabic. They first

run a phone recognizer on the speech input and use the resulting transcription to build a trigram language model. As we are dealing with written dialect data, only the second step is relevant to our work. Classification is done by minimizing the perplexity of the trigram models on the test segment.

An original approach to the identification of Swiss German dialects has been taken by the *Chochichästli-Orakel*.¹ By specifying the pronunciation of ten predefined phonetic and lexical cues, this web site creates a probability map that shows the likelihood of these pronunciations in the Swiss German dialect area. Our model is heavily inspired by this work, but extends the set of cues to the entire lexicon.

Computational methods are also used in dialectometry to assess differences between dialects with objective numerical measures. The most practical approach is to compare words of different dialects with edit distance metrics (Nerbonne and Heeringa, 2001). On the basis of these distance data, dialects can be classified with clustering methods. While the Swiss German data described here provide a valid base for dialect classification, this task is not the object of this paper.

3 Swiss German dialects

The German-speaking area of Switzerland encompasses the Northeastern two thirds of the Swiss territory. Likewise, about two thirds of the Swiss population define (any variety of) German as their first language.

It is usually admitted that the sociolinguistic configuration of German-speaking Switzerland is a model case of diglossia, i.e. an environment in which two linguistic varieties are used complementarily in functionally different contexts. In German-speaking Switzerland, dialects are used in speech, while Standard German is used nearly exclusively in written contexts.

Despite the preference for spoken dialect use, written dialect use has become popular in electronic media like blogs, SMS, e-mail and chatrooms. The Alemannic Wikipedia² contains about 6000 articles, among which many are written in a Swiss German dialect. However, all this data is very het-

¹<http://dialects.from.ch>

²<http://als.wikipedia.org>; besides Swiss German, the Alemannic dialect group encompasses Alsatian, South-West German Alemannic and Vorarlberg dialects of Austria.

Standard German	Swiss German	Validity Region	Example
<i>u</i>	<i>ue</i>	all	<i>gut</i> → <i>guet</i> ‘good’
<i>au</i>	<i>uu</i> [u:]	except Unterwalden	<i>Haus</i> → <i>Huus</i> ‘house’
<i>u</i>	<i>ü</i>	South (Alpine)	(<i>Haus</i> →) <i>Huus</i> → <i>Hüüs</i>
<i>ü</i>	<i>i</i>	South (Alpine), Basel	<i>müssen</i> → <i>miesse</i> ‘must’
<i>k</i> (word-initial)	<i>ch</i> [χ]	except Basel, Graubünden	<i>Kind</i> → <i>Chind</i> ‘child’
<i>l</i>	<i>u</i>	Bern	<i>alt</i> → <i>aut</i> ‘old’
<i>nd</i> (word-final)	<i>ng</i> [ŋ]	Bern	<i>Hund</i> → <i>Hung</i> ‘dog’

Table 1: Phonetic transformations occurring in Swiss German dialects. The first column specifies the Standard German graphemes. The second column presents one possible outcome in Swiss German; the area of validity of that outcome is specified in the third column. An example is given in the fourth column.

erogeneous in terms of the dialects used, spelling conventions and genres. Moreover, parallel corpora are virtually non-existent because need for translation is weak in a diglossic society.

The classification of Swiss German dialects is commonly based on administrative and topographical criteria. Although these non-linguistic borders have influenced dialects to various degrees, the resulting classification does not always match the linguistic reality. Our model does not presuppose any dialect classification. We conceive of the Swiss German dialect area as a continuum in which certain phenomena show more clear-cut borders than others. The nature of dialect borders is to be inferred from the data.³

Swiss German has been subject to dialectological research since the beginning of the 20th century. One of the major contributions is the *Sprachatlas der deutschen Schweiz* (SDS), a linguistic atlas that covers phonetic, morphological and lexical differences. Data collection and publication were carried out between 1939 and 1997 (Hotzenköcherle et al., 1962 1997). The lack of syntactic data in the SDS has led to a follow-up project called *Syntaktischer Atlas der deutschen Schweiz* (SADS), whose results are soon to be published (Bucheli and Glaser, 2002). Besides these large-scale projects, there also exist grammars and lexicons for specific dialects, as well as general presentations of Swiss German.

Swiss German dialects differ in many ways from Standard German. In the following sections, some of the differences in phonetics, lexicon, morphology and syntax are presented.

³Nonetheless, we will refer to political entities for convenience when describing interdialectal differences in the following sections of this paper.

3.1 Phonetic dialect differences

Table 1 shows some of the most frequent phonetic transformations occurring in Swiss German dialects. Note that our system applies to written representations of dialect according to the Dieth spelling conventions (Dieth, 1986). As a consequence, the examples are based on written dialect representations, with IPA symbols added for convenience in ambiguous cases. The Dieth rules are characterized by a transparent grapheme-phone correspondence and are generally quite well respected – implicitly or explicitly – by dialect writers.

The SDS contains two volumes of phonetic data, amounting to about 400 maps.

3.2 Lexical dialect differences

Some differences at the word level cannot be accounted for by pure phonetic alternations. One reason are idiosyncrasies in the phonetic evolution of high frequency words (e.g. *und* ‘and’ is reduced to *u* in Bern dialect, where the phonetic rules would rather suggest **ung*). Another reason is the use of different lexemes altogether (e.g. *immer* ‘always’ corresponds to *geng*, *immer*, or *all*, depending on the dialect).

The SDS contains five volumes of lexical data, although large parts of it concern aspects of rural life of the 1940s-1950s and are thus becoming obsolete. The *Wörterbuch der schweizerdeutschen Sprache*⁴ contains a much broader spectrum of lexical data, but its contents are difficult to access. Word lists published on the internet by dialect en-

⁴The *Wörterbuch der schweizerdeutschen Sprache* is a major lexicographic research project (Staub et al., 1881). Work started in 1881 and is scheduled to be fully achieved by 2020. Unfortunately, most of this work is not available in digital format, nor with precise geographical references. These issues are currently being addressed for the Austrian dialect lexicon in the project *dbo@ema* (Wandl-Vogt, 2008).

	1st Pl.	2nd Pl.	3rd Pl.
Standard	-en	-t	-en
West	-e	-et	-e
Wallis	-e	-et	-end, -und
East	-ed	-ed	-ed
Central	-id	-id	-id
Graubünden	-end	-end	-end

Table 2: Indicative plural suffixes of regular verbs in different Swiss German dialects. The first row shows the Standard German endings for comparison.

thusiasts certainly offer smaller coverage and lower quality, but can present an interesting alternative to extend lexical coverage.

3.3 Morphological and morphosyntactic dialect differences

Swiss German inflectional paradigms are generally reduced with respect to Standard German. Translation into Swiss German requires thus a set of morphosyntactic rules that insert, remove or reorder words in a sentence. For example, the lack of preterite tense in Swiss German requires all preterite sentences to be restructured as present perfect sentences. Similarly, the lack of genitive case gives rise to different syntactic structures to express possession. In contrast, Swiss German has clitic and non-clitic pronouns, a distinction that is not made in written Standard German.

On a purely morphological level, one can mention the verb plural suffixes, which offer surprisingly rich (and diachronically stable) interdialectal variation, as illustrated in Table 2. Minor interdialectal differences also exist in noun and adjective inflection.

In derivational morphology, the most salient dialect difference concerns diminutive suffixes: Swiss German has *-li* (or *-ji / -i* in Wallis dialect) instead of Standard German *-chen* and *-lein*.

Volume 3 of the SDS deals with morphology in the form of about 250 maps. Many morphosyntactic features of Swiss German are also investigated in the SADS survey.

4 Georeferenced transfer rules

The system proposed here contains sets of phonetic, lexical, morphological rules as illustrated in the examples above. Some of these rules apply uniformly to all Swiss German dialects, but most

of them yield different outcomes (variants) in different dialect regions. For example, the phonetic rule governing the transformation of word-final *-nd* will have four distinct variants *-nd*, *-ng*, *-n*, *-nt* (the *-nd* variant has been mentioned in Table 1). Each variant is linked to a probability map that specifies the areas of its validity. We refer to a rule, its associated variants and probability maps as a *georeferenced transfer rule*.

The maps for the georeferenced rules are extracted from the SDS. Currently, the system contains about 100 phonetic rules based on about 50 SDS maps. This corresponds to a fairly complete coverage. Lexical rules are currently limited to some high-frequency function words that are referenced in the SDS (about 100 rules). Morphological coverage is complete for regular inflection patterns and corresponds to about 60 rules. Some morphosyntactic and syntactic rules using unpublished SADS material have been added for testing purposes, but coverage is so far very limited.

4.1 Map generation

The SDS consists of hand-drawn maps on which different symbols represent different dialectal variants. Figure 1 shows an example of an original SDS map.

In a first preprocessing step, the hand-drawn map is digitized manually with the help of a geographical information system. The result is shown in Figure 2. To speed up this process, variants that are used in less than ten inquiry points are omitted. This can be justified by the observation by Christen (1998) that many small-scale variants in verbal morphology have disappeared since the data collection of the SDS in the 1940s and 1950s, while large-scale variants have not. We also collapse minor phonetic variants which cannot be distinguished in the Dieth spelling system.

The SDS maps, hand-drawn or digitized, are point maps. They only cover the inquiry points (about 600 in the case of the SDS), but do not provide information about the variants used in other locations. Therefore, a further preprocessing step interpolates the digitized point maps to obtain surface maps. We follow Rumpf et al. (2009) to create kernel density estimators for each variant. This method is less sensible to outliers than simpler linear interpolation methods. The resulting surface maps are then normalized such that at each point of the surface, the weights of all variants sum up to

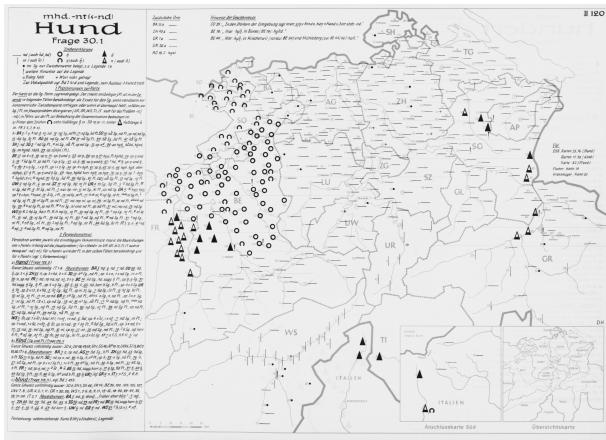


Figure 1: Original SDS map for the transformation of word-final *-nd*. The map contains four major linguistic variants, symbolized by horizontal lines (*-nd*), vertical lines (*-nt*), circles (*-ng*), and triangles (*-n*) respectively. Minor linguistic variants are symbolized by different types of circles and triangles.



Figure 2: Digitized version of the map in Figure 1.

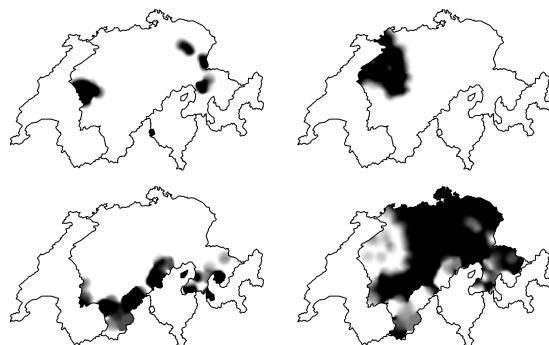


Figure 3: Interpolated surface maps for each variant of the map in Figure 2. Black areas represent a probability of 1, white areas a probability of 0.

1. These normalized weights can be interpreted as conditional probabilities $p(v | t)$, where v is a variant and t is the geographic location (represented as a pair of longitude and latitude coordinates). Figure 3 shows the resulting surface maps for each variant. Surface maps are generated with a resolution of one point per square kilometer.

Formally, the application of a rule is represented as follows:

$$R_{ij}(w_k) = w_{k+1}$$

where R_i represents the rule which addresses the i th phenomenon, and R_{ij} represents the j th variant of rule R_i . The result of applying R_{ij} to the word form w_k is w_{k+1} . The maps define probability distributions over rule variants at each geographic point t situated in German-speaking Switzerland (we call this set of points GSS), such that at any given point $t \in GSS$, the probabilities of all variants sum up to 1:

$$\forall i \quad \forall_{t \in GSS} \quad \sum_j p(R_{ij} | t) = 1$$

5 Three applications

The phonetic, lexical and morphological rules presented above allow to transform Standard German words into words of a specific Swiss German dialect. This rule base can be utilized in several NLP applications. The following sections will discuss the three tasks machine translation, dialect identification and dialect parsing.

5.1 Machine translation

Machine translation of a Standard German sentence begins with a syntactic and morphological analysis. Every word of the sentence is lemmatized (including compound word splitting), part-of-speech tagged and annotated with morphological features. The goal of this preprocessing is to take advantage of existing Standard German analysis tools to reduce ambiguity and to resolve some specific issues of German grammar like noun composition.⁵

Then, each annotated word is translated. Starting with the base form of the Standard German word, lexical rules are used to build a new Swiss German base form. If no lexical rule applies, the phonetic rules are used instead.

⁵For the time being, we perform this analysis simply by looking up word forms in a Standard German lexicon extracted from the Tiger treebank. Work is underway to merge the output of parsers like BitPar (Schmid, 2004) or Fips (Wehrli, 2007), part-of-speech taggers like TnT (Brants, 2000), and morphological analyzers like Morphisto (Zielinski and Simon, 2008) in order to provide accurate and complete annotation.

For example, the Standard German word *nichts* ‘nothing’ triggers a lexical rule; one variant of this rule, valid in the Northeast, yields the form *nünt*. In contrast, no lexical rule applies to the Standard German word *suchen*-VVFIN-3.Pl.Pres.Ind ‘they search’, which therefore triggers the following phonetic rules in Graubünden dialect:

- *u* → *u* (not *ü*)
- *u* → *ue*
- *e* → *a* (in diphthong)

and results in the stem *suach-*.

The georeferenced morphological rules represent a morphological generator for Swiss German: given a Swiss German base form and a set of morphological features, it creates an inflected dialectal form. In the above example, the Graubünden dialect suffix *-end* is attached, resulting in the inflected form *suachend*.

This approach of analyzing and recreating word forms may sound overly complicated, but allows generalization to (the morphological part of) morpho-syntactic restructuring like the transformation of preterite tense verbs into past participles. Similarly, it is easy to account for the fact that more Swiss German nouns build their plural with an *umlaut* than in Standard German.

The target dialect is fixed by the user by selecting the coordinates of a point *t* situated in German-speaking Switzerland.⁶ As illustrated above, the rules are applied sequentially, such that a Standard German word *w₀* yields an intermediate form *w₁* after the first transformation, and the final Swiss German form *w_n* after *n* transformations.

The probability resulting from the application of one rule variant *R_{ij}* transforming string *w_k* to *w_{k+1}* is read off the associated variant map at that point *t*:

$$p(w_k \rightarrow w_{k+1} | t) = p(R_{ij} | t) \quad s.t. \quad w_{k+1} = R_{ij}(w_k)$$

A derivation from *w₀* to *w_n*, using *n* transfer rules, yields the following probability:

$$p(w_0 \xrightarrow{*} w_n | t) = \prod_{k=0}^{n-1} p(w_k \rightarrow w_{k+1} | t)$$

The number *n* of rules in a derivation is not known in advance and depends on the structure of the word.

⁶Points are specified in the Swiss Coordinate System, either numerically or through a web interface based on Google Maps. The *nichts* example above assumed a point in the Northeast, while the *suchen* example assumed a point in the Southeast (Graubünden).

Note however that in transition zones, several variants of the same rule may apply. All rule applications are thus potentially ambiguous and lead to multiple derivations.⁷ Among multiple derivations, we choose the one that maximizes the probability.

The translation model presented here does not account for morphosyntactic adaptations and word reordering. While this word-by-word approach is sufficient in many cases, there are some important (morpho-)syntactic differences between Standard German and Swiss German (see section 3.3). Therefore, additional syntactic rules will provide context-dependent morphological and phonetic adaptations as well as word reordering in future versions of our system.

5.2 Dialect identification

Dialect identification or, more generally, language identification is commonly based on distributions of letters or letter n-grams. While these approaches have worked very well for many languages, they may be unable to distinguish related dialects with very similar phoneme and grapheme inventories. Moreover, they require training corpora for all dialects, which may not be available.

As an alternative, we propose to identify entire words in a text and find out in which regions these particular forms occur. This approach is similar to the *Chochichästli-Orakel*, but instead of using a small predefined set of cues, we consider as cues all dialect words that can be generated from Standard German words with the help of the transfer rules presented above. To do this, we first generate a list of Swiss German word forms, and then match the words occurring in the test segment with this list.

We obtained a list of lemmatized and morphologically annotated Standard German words by extracting all leaf nodes of the Tiger Treebank (Brants et al., 2002). Word forms that appeared only once in the corpus were eliminated. These Standard German words were then translated with our system. In contrast to the machine translation task, the target dialect was not specified. All potentially occurring dialect forms were generated and stored together with their validity maps.

For example, the *suchen* example yielded one single form *suachend* when restricted to a point in the Graubünden dialect area (for the translation

⁷We did not encounter cases where multiple derivations lead from the same Standard German word to the same Swiss German word. In that case, we would have to sum the probabilities of the different derivations.

task), but 27 forms when the target dialect was not specified (for the dialect identification task).

At test time, the test segment is tokenized, and each word of the segment is looked up in the Swiss German lexicon. (If the lookup fails, the word is skipped.) We then produce a probability map of each Swiss German word w_n by pointwise multiplication of all variant maps that contributed to generating it from Standard German word w_0 , in the same way as in the machine translation task illustrated above.

Note that a dialect form can be the result of more than one derivation. For example, the three derivations $sind\text{-VAFIN} \xrightarrow{*} si$ (valid only in Western dialects), $sein\text{-PPOSAT} \xrightarrow{*} si$ (in Western and Central dialects), and $sie\text{-PPER} \xrightarrow{*} si$ (in the majority of Swiss German dialects) lead to the same dialectal form si . In these cases, we take the pointwise maximum probability of all derivations $D(w)$ that lead to a Swiss German word form w :

$$\forall_{t \in GSS} p(w | t) = \max_{d \in D(w)} p(d | t)$$

Once we have obtained a map for each word of the segment, we merge them according to the following formula: The probability map of a segment s corresponds to the pointwise average of the probabilities of the words w contained in the sequence:

$$p(s | t) = \frac{\sum_{w \in s} p(w | t)}{|s|}$$

This is thus essentially a bag-of-words approach to dialect identification that does not include any notion of syntax.

5.3 Dialect parsing

A multidialectal parser can be defined in the following way: a source text, not annotated with its dialect, is to be analyzed syntactically. The goal is to jointly optimize the quality of the syntactic analysis and the dialect region the text comes from.

The exact implementation of dialect parsing is an object of future research. However, some key elements of this approach can already be specified.

Constituent parsers commonly consist of a grammar and of a lexicon. In a multidialectal parsing setting, the grammar rules as well as the lexicon entries have to be linked to probability maps that specify their area of validity. The lexicon built for the dialect identification task can be reused for parsing without further modifications. For the grammar however, more work is needed. A Swiss German

	Word-based	Trigram
Paragraphs (Wikipedia)	52.2%	86.7%
Sentences (Wikipedia)	31.3%	67.8%
Sentences (Non-Wiki.)	41.4%	44.4%

Table 3: F-measure values averaged over all six dialects.

grammar can be built by extracting a Standard German grammar from a treebank and manually modifying it to match the syntactic particularities of Swiss German (Chiang et al., 2006). In this process, the syntactic machine translation rules may serve as a guideline.

Instead of directly annotating each syntactic rule with a dialect parameter (Vaillant, 2008), we indirectly annotate it with a map containing its probability distribution over the dialect area.

6 Evaluation

6.1 Dialect identification

In terms of annotated resources for evaluation, dialect identification is the least demanding task: it requires texts that are annotated with their respective dialect. Such data can be extracted from the Alemannic Wikipedia, where many Swiss German articles are annotated with their author’s dialect.

We extracted about ten paragraphs of text for six dialect regions: Basel, Bern, Eastern Switzerland, Fribourg, Wallis and Zurich. The paragraphs amount to a total of 100 sentences per region.⁸ The surfaces of these six regions were defined using political (canton) boundaries and the German-French language border.

The dialect identification system scored each paragraph s with a probability map. We calculated the average probability value for each of the six regions and annotated the paragraph with the region obtaining the highest value:

$$Region(s) = \arg \max_{Region} \left(\frac{\sum_{t \in Region} p(s | t)}{|Region|} \right)$$

We tested entire paragraphs and single sentences, and repeated both experiments with a simple trigram model trained on Wikipedia data of similar size. The results of these tests are summarized in Table 3 (first two rows).

⁸The choice of the dialects and the size of the corpus was largely determined by the data available. The average sentence length was 17.8 words per sentence.

We suspected that the outstanding results of the trigram model were due to some kind of overfitting. It turned out that the number of Swiss German Wikipedia authors is very low (typically, one or two active writers per dialect), and that every author uses distinctive spelling conventions and writes about specific subjects. For instance, most Zurich German articles are about Swiss politicians, while many Eastern Swiss German articles are about religious subjects. Our hypothesis was thus that the n-gram model learned to recognize a specific author and/or topic rather than a dialect.

In order to confirm this hypothesis, we collected another small data set from various web resources (not from Wikipedia, 50 sentences per dialect).⁹ Table 3 (last row) indeed confirms our suspicion. The performance of the trigram model dropped by more than 20 percent (absolute), while the word-based model surprisingly performed better on the second test set than on the Wikipedia data. One possible explanation is the influence of Standard German spelling on the Wikipedia data, given that many Swiss German articles are translations of their Standard German counterparts. However, we have not thoroughly verified this claim.

While our dialect identification model does not outperform the trigram model, recent adaptations show promising results. First, the dialect annotation based on average probability values penalizes large and heterogeneous regions, where a high-probability sub-region would be cancelled out by a low-probability sub-region. Using maximum instead of average could improve the dialect annotation. Second, not all derivations are equally relevant; for example, word frequency information can provide a crucial clue to weighting derivations.

6.2 Machine translation and parsing

For the evaluation of the machine translation task, we might again resort to data from Wikipedia. As said above, many articles are translations from Standard German and can serve as a small parallel (or at least comparable) corpus. In addition, we plan to extract Swiss German text from other sources and have it translated into Standard German.

Current translation evaluation metrics like BLEU or TER only use binary measures of word match. Given the importance of phonetic transfor-

mations in our approach, and given the problems arising from lacking spelling conventions, finer-grained metrics might be needed in order to account for different degrees of word similarity.

While the machine translation system has not been evaluated yet, a prototype version is accessible on the Web.¹⁰

For parsing, the data requirements are even more demanding. Syntactically annotated Swiss German dialect texts do not currently exist to our knowledge, so that a small evaluation tree bank would have to be created from scratch.

7 Conclusion

We have presented an approach to natural language processing that takes into account the specificities of Swiss German dialects. Dialects have too often been viewed as homogeneous entities clearly distinguishable from neighbouring dialects. This assumption is difficult to maintain in many dialect areas. Rather, each dialect is defined as a unique combination of variants; some variants may be shared with adjacent dialects, others may act as discriminating features (isoglosses). Our approach reflects this point of view by modelling an entire dialect continuum.

The data for our model come from dialectological research. Dialects may be among the few language varieties where linguistically processed material is not significantly costlier to obtain than raw textual data. Indeed, data-driven approaches would have to deal with data sparseness and dialectal diversity at the same time. While processing dialectological data is tedious, we have proposed several tasks that allow the data to be reused.

This paper reflects the current status of ongoing work; while data collection is fairly complete, evaluating and tuning the proposed models will be a high priority in the near future.

Besides presenting a novel approach to NLP tasks, we argue that dialectological research can also profit from this work. Dialectological research has traditionally suffered from lack of dissemination among laymen: dialect atlases and lexicons are complex pieces of work and often difficult to access. Dynamic models of dialect use could bring dialectological research closer to a large audience, especially if they are freely accessible on the internet.

⁹The gold dialect of these texts could be identified through metadata (URL of the website, name and address of the author, etc.) in all but one case; this information was checked for plausibility by the authors.

¹⁰<http://lat1cui.unige.ch/~yves/>

Acknowledgements

Part of this work was carried out during the first author's stay at Columbia University, New York, funded by the Swiss National Science Foundation (grant PBGEP1-125929).

References

- Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken Arabic dialect identification using phonotactic modeling. In *Proceedings of the EACL'09 Workshop on Computational Approaches to Semitic Languages*, Athens.
- S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.
- Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of NAACL 2000*, Seattle, USA.
- Claudia Bucheli and Elvira Glaser. 2002. The Syntactic Atlas of Swiss German dialects: empirical and methodological problems. In Sjef Barbiers, Leonie Cornips, and Susanne van der Kleij, editors, *Syntactic Microvariation*, volume II. Meertens Institute Electronic Publications in Linguistics, Amsterdam.
- David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Saifullah Shareef. 2006. Parsing Arabic dialects. In *Proceedings of EACL'06*, pages 369–376, Trento.
- Helen Christen. 1998. *Dialekt im Alltag: eine empirische Untersuchung zur lokalen Komponente heutiger schweizerdeutscher Varietäten*. Niemeyer, Tübingen.
- Antonio M. Corbí-Bellot, Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor, and Kepa Sarasola. 2005. An open-source shallow-transfer machine translation engine for the Romance languages of Spain. In *Proceedings of EAMT'05*, pages 79–86, Budapest.
- Rodolfo Delmonte, Antonella Bristot, Sara Tonelli, and Emanuele Pianta. 2009. English/Veneto resource poor machine translation with STILVEN. In *Proceedings of ISMTCL*, volume 33 of *Bulag*, pages 82–89, Besançon.
- Eugen Dieth. 1986. *Schwyzerütschi Dialäktschrift*. Sauerländer, Aarau, 2 edition.
- Martin Forst. 2002. La traduction automatique dans le cadre formel de la LFG – Un système de traduction entre l'allemand standard et le zurichois. In *Publications du CTL*, volume 41. Université de Lausanne.
- Jan Hajic, Petr Homola, and Vladislav Kubon. 2003. A simple multilingual machine translation system. In *Proceedings of the Machine Translation Summit XI*, pages 157–164, New Orleans.
- Rudolf Hotzenköcherle, Robert Schläpfer, Rudolf Trüb, and Paul Zinsli, editors. 1962–1997. *Sprachatlas der deutschen Schweiz*. Francke, Bern.
- Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon*, pages 9–16, Philadelphia.
- Grzegorz Kondrak and Tarek Sherif. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of the ACL Workshop on Linguistic Distances*, pages 43–50, Sydney.
- Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL'01*, Pittsburgh.
- John Nerbonne and Wilbert Heeringa. 2001. Computational comparison and classification of dialects. In *Dialectologia et Geolinguistica. Journal of the International Society for Dialectology and Geolinguistics*, number 9, pages 69–83. Edizioni dell'Orso, Alessandria.
- Jonas Rumpf, Simon Pickl, Stephan Elspaß, Werner König, and Volker Schmidt. 2009. Structural analysis of dialect maps using methods from spatial statistics. *Zeitschrift für Dialektologie und Linguistik*, 76(3).
- Yves Scherrer. 2007. Adaptive string distance measures for bilingual dialect lexicon induction. In *Proceedings of the ACL'07 Student Research Workshop*, pages 55–60, Prague.
- Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of COLING'04*, Geneva, Switzerland.
- Friedrich Staub, Ludwig Tobler, Albert Bachmann, Otto Gröger, Hans Wanner, and Peter Dalcher, editors. 1881-. *Schweizerisches Idiotikon : Wörterbuch der schweizerdeutschen Sprache*. Huber, Frauenfeld.
- Pascal Vaillant. 2008. A layered grammar model: Using tree-adjoining grammars to build a common syntactic kernel for related dialects. In *Proceedings of TAG+9 2008*, pages 157–164, Tübingen.
- Eveline Wandl-Vogt. 2008. An der Schnittstelle von Dialektwörterbuch und Sprachatlas: Das Projekt “Datenbank der bairischen Mundarten in Österreich electronically mapped (dbo@ema)”. In Stephan Elspaß and Werner König, editors, *Germanistische Linguistik 190-191. Sprachgeographie digital. Die neue Generation der Sprachatlanten*, pages 197–212. Olms, Hildesheim.

Éric Wehrli. 2007. Fips, a “deep” linguistic multilingual parser. In *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, pages 120–127, Prague.

Andrea Zielinski and Christian Simon. 2008. Morphisto – an open-source morphological analyzer for German. In *Proceedings of FSMNLP’08*, Ispra, Italy.

SHORT PAPERS

Clustering Urdu verbs on the basis of related aspectual auxiliaries and light verbs

Tafseer Ahmed

Department of Linguistics

Universität Konstanz

Germany

tafseer@gmail.com

Abstract

The paper describes work done on Urdu aspectual auxiliaries/ light verbs. The frequencies of main_verb-auxiliary and main_verb-light_verb sequences are obtained by shallow processing of an Urdu corpus. The normalized data corresponding to each of the main verbs is used for clustering (unsupervised learning) of Urdu verbs. It gives four major clusters of verbs that are hierarchically arranged. The inspection of the verbs in these clusters show that most of the verbs in a cluster share some common semantic property and these clusters correspond to semantic classes of the verbs. Some other properties about auxiliary/light verbs are also found in this analysis.

1 Introduction

Urdu is an Indo-Aryan language spoken in Pakistan and India. It is closely related to Hindi with same grammatical structure but differences in script and vocabulary. In Urdu, we find sequences of verbs in which the main verb is followed by another verb (Schimdt, 1999). The following-verb can be an auxiliary, a modal or a light verb. Consider these examples.

- (1) TrEn A-I (thI)
train come-Perf.F.SG Past.F.SG
'The train came.'
- (2) TrEn A rah-I thI
train come Prog-F.SG Past.F.SG
'The train was coming.'
- (3) TrEn A ga-I (thI)
train come go-F.SG Past.F.SG
'The train had come.'

All of the above examples have tense auxiliaries at the last position. The tense auxiliary can follow the main verb or the verb-verb sequence. However, we are not interested in tense auxiliaries. In

this paper, we are interested in the verbs like *rah* 'stay' (used for progressive) and *jA* 'go' (use for completion).

Siddiqui (1971) and Hook (1974) provided a list of verbs that follow main verbs. The verbs are: *dE* 'give', *lE* 'take', *A* 'come', *jA* 'go', *DAl* 'insert', *paR* 'fall', *rah* 'stay', *beTH* 'sit', *cuk* (for completion), *sak* (for ability), *pA* 'get', *kar* 'do', *hO* 'be', *uTH* 'rise', *cAH* 'want', *dE* 'give', *rakH* 'put', *ban* 'get make', *lag* 'touch/hit', *nikal* 'come out', *Tahar* 'stop' and *cal* 'move'.

As mentioned earlier, this list does not contain a single type of verbs. The list includes auxiliaries e.g. *rah* for progressive, modals e.g. *cAH* for want and light verbs e.g. *jA* for completion. We present all of these in a single list, because in the latter part of this paper we argue that some of the auxiliaries especially the progressive auxiliary *rah* 'stay' are correlated to certain semantic classes of the verb. Hence, we need to study the behavior of all of these verbs irrespective of their syntactic properties. We use the term V2 for all of these verbs throughout this paper. Many writers e.g. Butt (1995) distinguish between aspectual auxiliaries and light verbs, but we use the same term V2 for all these verbs.

There is no significant work on the semantic verb classes of Urdu. There are few references to some verb classes such as ingestives (Saksena 1982) and the intransitive verbs that allow optional *ne* (Butt 1995).

The modals and aspectual auxiliaries can follow any verb, but it is not the case for light verbs. Consider the following example.

- (4) a. gARI cal dI
vehicle.F.SG move give.Perf.F.SG
'The vehicle started moving.'
- b. *gARI ruk dI
vehicle.F.SG stop give.Perf.F.SG
'The vehicle stopped.'

The light verb *dE* 'give' is not used with the verb *ruk* 'stop'. Hence, each light verb is compatible with some, and does not occur with the verbs of other semantic classes.

Our experiment tests the hypothesis and investigates whether there is a correlation between the progressive marker and certain verb class(es).

There is an important syntactic issue in the processing of V2 verbs. Each V2 verb governs the morphological form of the main verb preceding it. Different morphological forms of the previous verb correspond to different syntactic/semantic interpretation of the V2 following it.

Consider the example of *jA* 'go'. It can be interpreted as passive marker, completion marker or continuity marker on the basis of the form of the main verb preceding it. If the (preceding) main verb is in perfective form, *jA* is considered as the passive marker. If the main verb is in root form, *jA* is considered as the completion marker and if the imperfective form of the main verb is used, it is considered as a continuity marker. See the following examples.

- (5) sEb kHA-yA
apple.M.Sg eat-Perf.M.Sf
 gayA
 go.Perf.M.Sg
'Apple was eaten.' (Passive)
- (6) sEb pak gayA
apple.M.Sg ripe go.Perf.M.Sg
'Apple had ripen.' (Completion)
- (7) vuH sEb kHA-tA
3SG apple eat-Impf.M.Sg
 gayA
 go.Perf.M.Sg
'He kept on eating the apples.' (Continuity)

The aim of our experiment is to analyze the corpus to get some empirical results about V2 verbs and the related issues introduced above. What is the behavior of these V2 verbs related to different verbs and different classes of verbs? Can we find verb classes based on distribution of V2 verbs with the main verb?¹

¹There are certainly other features like subcategorization frame and alternations that can be used in verb clustering, however we tried to find out how much can be done solely on the basis of these (V2) verbs.

2 Experiment

In the above section, we presented a hypothesis that there is a semantic relation among many of V2 verbs and the main verbs. In this experiment, we try to find empirical evidence for this hypothesis. The experiment has two parts. The first part provides the frequency of each V2 verb corresponding to each main verb. These frequencies can tell us about the syntactic/semantic properties of V2 verbs. The second part of the experiment tries to cluster the (main) verbs on the basis of frequencies of V2 verbs associated with them. Can we find semantic classes on the basis of V2 verb frequencies?

It is not easy (and possible with limited resources) to employ deep parsing methods to perform this experiment. For Urdu, there is no tree bank available. Similarly, no sizable POS tagged corpus is available. Moreover, no morphological analyzer is publicly available that is easily integratable with other applications. Hence, it is necessary to use shallow methods to perform this experiment.

We plan to count the occurrence of the main verb followed by the V2 verb. The main verb can be in one of the different morphological forms as Urdu verb is inflected on the basis of number, gender and/or person agreement. As we are not able to use a morphological analyzer, we planned to obtain data only for those V2 verbs that are followed by the (uninflected) root form of the main verb.² There are 12 such V2 verbs that are preceded by the root form of the main verb. We use these verbs in our experiment. The list of these verbs is present in table 1.

A list of Urdu (main) verbs is obtained from Humayoun's (2006) online resources. As most of the Urdu verbs are in form of noun/adjective + verb complex predicate e.g. *intizAr* 'wait' *kar* 'do' (for 'wait'), there are less than thousand simple verbs e.g. *gir* 'fall' in Urdu. The used verb list contains these simple verbs only.

There is a potential problem in using the root form of main verb without deep processing. The masculine singular perfective form of a verb is form identical to its root causative form. For example, the verb *gir* '(get) fall' has perfective form *girA* used for masculine singular agreement. The root form of the corresponding causative verb is *girA* '(make) fall'. (The perfective form of this causative

²The native speaker knowledge tells that these V2 verbs are most frequently used in Urdu. So, it can be assumed that we do not lose much data.

verb *girA* is *girAyA* for masculine singular agreement.) Hence, we remove all such verb-causative pairs that introduce this form ambiguity.

As we use the V2 verbs that are preceded by the root form of the main verb, we do not need to search the other morphological forms of the main verb. However, we do need to find different morphological forms of V2 verbs immediately following the root form of the main verbs. For this purpose, we manually generated all the morphological forms of these twelve V2 verbs.

As a corpus, we processed 7337 documents having 14,196,045 tokens. The documents are obtained from CRULP's (www.crulp.org) Urdu corpus and websites www.urduweb.org and www.kitaabghar.com.

The documents of the corpus are processed one by one. The text of each document is divided into sentences on the basis of sentence breakers. Each word of these sentences is matched with the list of the main verbs. If the word is found in Urdu verb list, the next word is matched with the (inflected) words in the V2 verb list. If it is also found, we increase the count of that verb-V2 combination. To make the data better for normalization, the count of each main verb in imperfective form is also calculated.

After the processing of all the documents of the corpus, we got a table having counts of verb-V2 combinations. We selected 183 verbs for further processing. These are the verbs for which the sum of all the counts is greater than 20.

These data is to be normalized for further processing. The count of each verb-V2 combination is divided by the sum of counts of all combinations for that verb (plus counts of imperfective forms). This gives normalized frequencies of the combination that can be compared in further processing. The normalized frequency table for some verbs is given in table 1. As the sum at denominator includes the count of imperfective forms, the frequencies in each column (that use V2 counts only) do not add up to 1.

The normalized frequencies for the combinations corresponding to each main verb constitute a vector. These vectors are used for clustering that is the unsupervised learning of classes. The software tool Cluster 3.0 is used for hierachal cluster of these vectors using centroid method. The tool is available at <http://bonsai.ims.utkyo.ac.jp/mdehoon/software/cluster/software.htm>.

V2/main	<i>gir</i> 'fall'	<i>hans</i> 'laugh'	<i>tOR</i> 'break'
<i>rah</i> 'stay'	0.0937	0.1064	0.0771
<i>dE</i> 'give'	0.0032	0.0292	0.5176
<i>lE</i> 'take'	0	0.0133	0.0193
<i>A</i> 'come'	0.0032	0	0
<i>jA</i> 'go'	0.4345	0	0.1350
<i>DAI</i> 'insert'	0	0	0.0354
<i>paR</i> 'fall'	0.1260	0.1064	0
<i>bETH</i> 'sit'	0.0016	0	0
<i>cuk</i> complete	0.0339	0	0.0354
<i>sak</i> able	0.0129	0.0026	0.0482
<i>pA</i> 'find'	0.0016	0	0
<i>uTH</i> 'rise'	0	0	0

Table 1: A sample from the Frequency table corresponding to main_verb-V2 sequences

3 Results

The hierachal clusters obtained by this exercise are shown in figure 1.

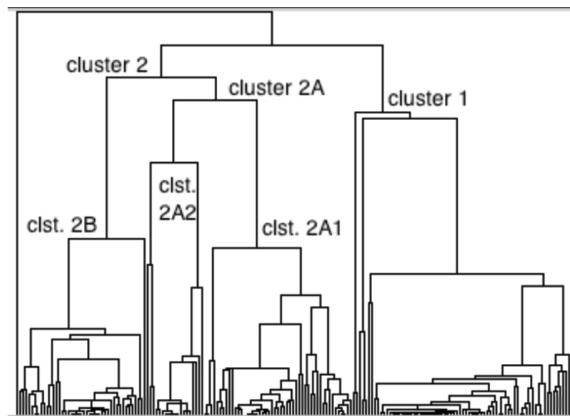


Figure 1: The dendrogram showing hierachal clustering of Urdu verbs on the basis of V2 frequencies.

Visually we can see two/four major clusters. There are two major clusters: cluster 1 and cluster 2. Cluster 2 is subdivided into two major clusters: 2A and 2B. One of these two clusters i.e. 2A is subdivided into two more clusters 2A1 and 2A2. Hence, visually we find four major clusters: 1, 2A1, 2A2 and 2B that are hierarchically arranged. Some of the verbs from these clusters are:

Cluster 1: (approx. 70 verbs) *nikal* 'emerge/come out', *jI* 'live', *A* 'come', *jal* '(get) burn', *guzar* 'pass', *pak* '(get) bake', *ban* '(get) make', *tHak* 'get tired', *kUd* 'jump (from one point to other)', *ucHal* 'jump (up and down motion)

kHA 'eat', *samajH* 'understand', *jIt* 'win', *lUT* 'rob', *nahA* 'bath', *pI* 'drink', *nigal* 'swallow'

Cluster 2A1: (approx. 45 verbs) *cal* 'move', *hans* 'laugh', *gA* 'sing', *muskurA* 'smile', *bOl* 'speak', *kHEL* 'play', *cIx* 'scream', *nAc* 'dance', *laTak* '(get) hang', *jAg* 'wake up'

paRH 'read', *dEkH* 'see', *sun* 'hear', *mAng* 'ask', *cUs* 'suck', *pIT* 'hit/beat', *kamA* 'earn', *lA* 'bring'

pUcH 'ask', *DHUND* 'search', *cHU* 'touch', *kANp* 'shiver', *baj* '(get) ring'

Cluster 2A2: (approx. 20 verbs) *jAn* 'know', *pahcAn* 'recognize', *apnA* 'adopt', *pahan* 'wear', *mAn* 'accept', *tHAM* 'hold', *kHENc* 'pull', *cun* 'pick/pluck'

gin 'count', *dHO* 'wash', *pIs* 'grind', *cHAn* 'filter', *gHER* 'cover', *tal* 'fry'

Cluster 2B: (approx. 40 verbs) *kah* 'say', *dE* 'give', *likH* 'write', *bEc* 'sell', *sukHA* '(make) dry', *navAz* 'give', *batA* 'tell', *jagA* '(make) wake up', *toR* 'break', *kHol* 'open', *rakH* 'put', *rOK* '(make) stop', *bAnT* 'divide/distribute', *kas* 'tighten'

4 Discussion

These clusters correspond to semantic classes discussed earlier in the literature. Most of the verbs in the cluster 1 are unaccusative verbs whose subject is a patient/theme. These unaccusative verbs e.g. *nikal* 'emege/come-out' etc. are listed in first paragraph of cluster 1 verb list given in the above section. The second paragraph in this list has another class of verbs i.e. ingestives. The verbs like *kHA* 'eat' etc. are transitive. However, the subject of these verbs is considered as a theme that traverses a path (the object) (Ramchand 2008). Hence ingestives are semantically closer to the unaccusatives. The subjects of both are patient/theme or undergoer in Ramchand's framework.

Cluster 2 corresponds to the (transitive and intransitive) verbs that have agentive subjects. The cluster 2B corresponds to the transitive verbs whose subject bring some change to the object. Most of the verbs are either causing change of state verbs like *toR* 'break' or *bHigO* '(make) wet' or ditransitives like *kah* 'say' or *rakH* 'put'. The subject does not get affected in these types of verbs.

The analysis of verb list of cluster 2A1 shows that it has three kinds of verbs (listed in three paragraphs in above section). The verbs in first paragraph e.g. *cal* 'move' and *hans* 'laugh' etc. correspond to the (intransitive) unergative verbs that have agentive subject. Most of the verbs in second

and third paragraphs e.g. *dEkH* 'see'/*kamA* 'earn' and *pUcH* 'ask'/*DHUND* 'search' are transitive verbs whose subject is agentive.

Most of the verbs in class 2A2 are those whose subject gets something physically or logically. The verbs e.g. *cun* 'pick' in the first paragraph easily fit this description, however the verbs in second paragraph form a pragmatic class of those actions e.g. *pIs* 'grind' in which the subject often gets benefit logically.

It must be noted that the "verbs in nth paragraphs" described in above text are not the sub-cluster given by the clustering tool. We subjectively (and manually) made these subdivisions among the verbs of each clusters to adequately explain the semantics of the verbs in each cluster.

When we sort the frequency table (having cluster labels) with respect to frequencies of V2 verbs, we find interesting observations. The frequencies of progressive auxiliary *rah* 'stay' is correlated to the cluster 2A1. It means that this auxiliary occurs with all kinds of verbs, but it is more frequent with verbs of certain semantic properties. However, the high frequency occurrences of the verb *sak* used as ability marker do not correlate to any verb class.

The frequency analysis gives the productivity/compatibility of each of the V2 verbs. The progressive marker *rah* is found to occur with 161 (out of 183) verbs. The light verb *jA* is found to occur with 121 verbs. On the other hand, light verbs *DAI* 'put' and *uTH* 'rise' are found to appear only with 22 and 23 (main) verbs respectively.

5 Conclusion

Urdu corpus is processed to find frequencies of main_verb-V2(auxiliary/light_verb) combinations. The clustering of this data gives four major clusters that have verbs with common semantic properties. This experiment is an effort to find how much we can comprehend about semantics of Urdu verbs solely on the basis of light verbs/auxiliary frequencies.

References

- Peter Edwin Hook. 1974. *The Compound Verbs in Hindi*. University of Michigan, Ann Arbor.
- Muhammad Humayoun. 2006. Urdu morphology, orthography and lexicon extraction. Masters thesis, Chalmers University of Technology, Chambéry.

Gillian Ramchand. 2008. *Verb Meaning and the Lexicon: A First Phase Syntax*. Cambridge University Press, Cambridge.

Anuradaha Saksena. 1982. *Topics in the Analysis of Causatives with an Account of Hindi Paradigms*. University of California Press, Berkeley.

Ruth Laila Schimdt. 1999. *Urdu: An Essential Grammar*. Routledge, London.

Abul Lais Siddiqui. 1971. *Jamaul Qawaaid (Comprehensive Grammar)*. Karachi.

Qualitative and Quantitative Error Analysis in Context

Christian Chiarcos, Julia Ritz

Collaborative Research Centre 632 ‘Information Structure’

Universität Potsdam

Potsdam, Germany

{chiarcos, jritz}@uni-potsdam.de

Abstract

A crucial step in the development of NLP systems is a detailed error analysis. Our system demonstration presents the infrastructure and the workflow for training classifiers for different NLP tasks and the verification of their predictions on annotated corpora. We describe an enhancement cycle of subsequent steps of classification and context-sensitive, qualitative error analysis using ANNIS2 (Chiarcos et al., 2009), an annotation information system for querying and visualizing corpora with multi-layer annotations.

We illustrate our approach for the example of a classifier that distinguishes anaphoric and non-anaphoric NPs. Our implementation is, however, not specific to this particular problem, but rather, it makes use of generic modules suitable to represent and process any graph-based linguistic annotations. The workflow described here can thus be employed to different NLP tasks, ranging from simple part-of-speech tagging to coreference analysis or semantic role labelling.

1 Background

The training of statistical classifiers for NLP tasks requires a well-informed selection of parameters, as well as a detailed error analysis for their verification and adjustment, and here, we show how a general-purpose corpus query tool like ANNIS2 (Chiarcos et al., 2009) can be applied in such a cycle of classifier refinement and qualitative and quantitative error analysis. Already Chiarcos et al. (2009) sketched the possibility of such a workflow, but in this paper we explore the potential of such an application in depth for a concrete example, the training of a classifier for NP anaphoricity (Ritz, 2010) using WEKA (Witten and Frank, 2005).

As opposed to earlier research in the field of annotation enrichment and classifier refinement, ANNIS2 is not tied to one particular type of annotation: It allows to store, visualize and query any kind of text-based linguistic annotations (see fn. 2), with annotations representing plain text, hyper references, but also multi-media material such as sound files and video sequences. As a tool specifically designed for multi-layer annotations, it comes with the following advantages:

domain independence ANNIS2 can cope with any kind of text-based annotation, so it allows to integrate multiple layers of annotation regardless of compatibility considerations. Existing frameworks that employ cyclic application of statistical classifiers and qualitative error analysis are usually domain-specific,¹ whereas ANNIS2 provides generic visualization components. With ANNIS2, different kinds of linguistic annotations can be inspected using a single, yet domain-independent tool.

elimination of feature extraction Taking the integration of syntactic and semantic features in a classifier for anaphor resolution as an example, it is necessary to represent such higher-level information as attribute-value pairs. Using ANNIS2, syntactic information can be directly drawn from the respective annotations, because all kinds of annotations and coreference are represented using the same graph-based data structures. Fea-

¹An active learning system for the acquisition of parser rules, for example, displays training sentences besides their meaning representation, cf., (Zelle and Mooney, 1996; Thompson et al., 1999), but cannot be easily extended to display, say, coreference. A system for the disambiguation between different candidate parses produced by the same tool (Osborne and Baldwin, 2004) is even more restricted to the specific syntactic formalism.

ture extraction can thus be directly performed by ANNIS2, without the need of error-prone conversion scripts mapping between two or more different formats that represent different layers of annotation.

multi-layer visualization ANNIS2 allows to visualize the results of the classifier, together with all other annotations, regardless of whether they represent features attached to single tokens or edge labels in a syntax tree. Thus, a qualitative analysis can be performed that allows to study the relationship between any layer of original annotation and the results of a classifier.

2 From Corpus Preparation to Classifier Refinement

We describe the application of the ANNIS data base for the development and the interactive refinement of a classifier that distinguishes anaphoric and non-anaphoric NPs on the the OntoNotes corpus (Hovy et al., 2006; Weischedel et al., 2007).

In its current instantiation ANNIS2, ANNIS is a relational database implementation that allows to store, to visualize and to query for any kind of text-oriented linguistic annotation using a web-based interface.² Basic data structures are **nodes** (tokens, spans of tokens) and **edges** (relations between nodes) with different subtypes that represent the basis for the query language and the generic visualizations of ANNIS (Chiarcos et al., 2009). ANNIS2 also provides the feature to export query matches to external tools such as WEKA, so that further statistical evaluations are possible, whose results can then be merged with the original annotations and reimported into ANNIS.

The overall workflow is shown in Fig. 1.

Corpus Preparation For our experiment, we imported the syntactic and the coreference annotations of the OntoNotes corpus into ANNIS2. To our best knowledge, ANNIS2 is the only graphical

²The underlying data structures of ANNIS are specified in the PAULA Object Model (Chiarcos et al., 2009) as sets of labelled directed acyclic graphs (LDAGs). In agreement with Ide and Suderman (2007), we assume that practically every kind of linguistic annotation that refers to textual representations of natural languages can be modelled by means of LDAGs. Here, we describe here the study of interdependencies between syntactic annotation and coreference annotations. Other applications include the comparison between alternative parses for the same text (Chiarcos et al., 2010) or multi-layer annotations in general (Dipper and Götze, 2005).

tool available that allows for the concurrent study of two independent layers of annotation.³

The syntactic annotations and the coreference annotations are converted to PAULA, the data format underlying ANNIS (Dipper and Götze, 2005), by means of a set of converters provided over our web interface (upper left corner in Fig. 1).⁴ Using Kickstarter, a graphical user interface for the maintenance of an ANNIS2 installation, the resulting PAULA corpus is compiled to the relational DB format and then loaded into the data base (left side in Fig. 1).

The ANNIS2 screenshot in Fig. 1 shows the results for a query for two coreferent OntoNotes markables, the tree view of the syntax annotation, and a fraction of the pointing relation visualization with selected anaphoric chains being highlighted. The aforementioned WEKA export can be exploited to extract features from the annotations.

Feature Extraction The following ANNIS2 query extracts all anaphoric NPs:

```
(1) cat="NP" &
node #2 _=_ #1 &
node #2 ->IDENT_relation #3
```

The first line of (1) retrieves an NP from the syntax annotation and binds it to variable #1. Then, we query for a node #2 from the coreference annotation that covers the same tokens as #1 and that takes a third node #3 as its anaphoric (IDENT) antecedent. Node #1 is thus anaphoric.

ANNIS2 can export the matches for query (1) in ARFF, WEKA's native table format. In the ARFF table, then, every line represents a match for this query; every column represents an attribute assigned to the nodes #1, #2 or #3; every cell represents the annotation value (or NULL if the attribute is not specified). The first two columns identify the anchor of the match, i.e., the document id and the id of node #1; when an ARFF table is merged with the original corpus, the classification results are represented as annotations attached to this anchor node.

³Although TrED (Pajas and Štěpánek, 2009) allows to represent and to query for anaphoric and syntactic relations, as well, it is limited to annotations that form one single tree, so that at most one layer of syntactic annotation is available, whereas ANNIS2 is unrestricted in this respect.

⁴All resources mentioned in this paper, ANNIS2, Kickstarter, and the web interface to converter and merging scripts can be found under our project site <http://www.sfb632.uni-potsdam.de/~d1>. For WEKA, see <http://www.cs.waikato.ac.nz/ml/weka>.

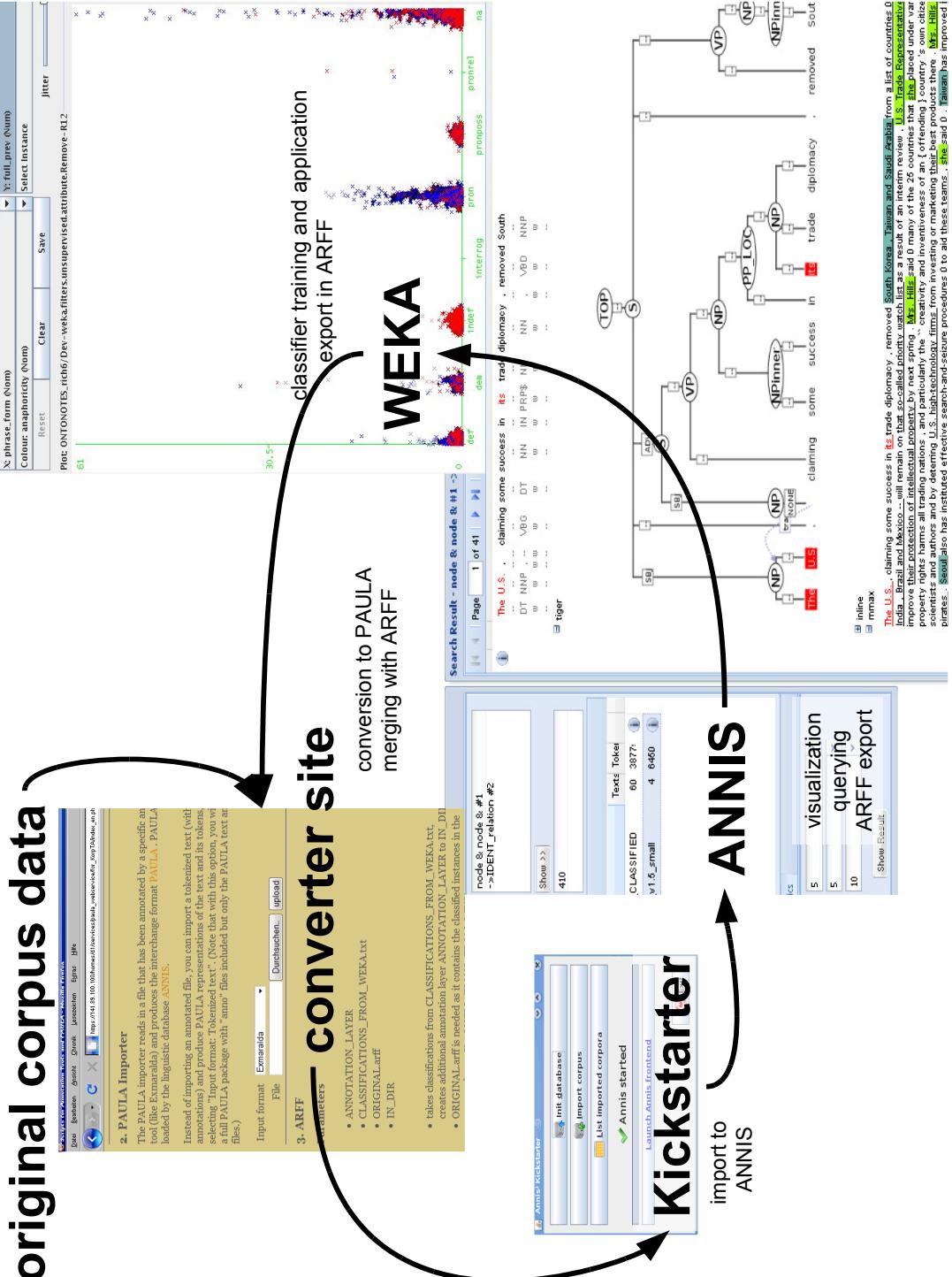


Figure 1: Workflow

To mark the matches of the query as anaphoric NPs, a new column `anaph` with value `yes` is added. Every NP not matching the query (i.e., non-anaphoric NPs) is thus characterized by `anaph=NULL`. These features define our target classification for anaphoricity.

As the ARFF file contains the original document and node ids of the anchor nodes, the ARFF table can be merged with the original corpus. For this purpose, the converter site provides a CGI script that takes the text file, the ARFF file, and an archive comprising the original corpus as its parameters. In this way, users can enrich their original corpora with WEKA classification results by means of a web interface. The modified corpus is then imported into ANNIS2 using Kickstarter.

Retrieving Training Data and Classifier Training A query for all NPs (`cat="NP"`) now yields not only all annotations originally assigned to the corresponding node #1, but also, a column for `anaph` which is `yes` for all nodes retrieved by query (1) before, and `NULL` for all other nodes. The attribute `anaph` is unspecified (`NULL`) for all generic, underspecified, abstract, attributive and appositive NPs that are hence considered non-anaphoric.

The ARFF table exported from ANNIS2 can directly be loaded into WEKA. In Fig. 1, upper right corner, we see a WEKA plot of three parameters, the phrase form (type of referring expression, X), the number of previous mentions of the same expression (Y), and the anaphoricity of the respective expression (color: non-anaphoric as red, anaphoric as blue). As the plot shows, pronouns are mostly anaphoric (blue), indefinites are non-anaphoric (red). Possessive NPs are *mostly* non-anaphoric, but definites cannot be directly associated with either anaphoric or non-anaphoric. This indicates that the surface form provides us at best with an incomplete approximation of anaphoricity.

Extending the set of factors, however, allows us to develop a more reliable classifier. We trained C4.5 decision trees on a training set consisting of 80% of the documents in the corpus, and applied to a development consisting of another 10% of the corpus' documents. Details of the classifier and its performance are described by Ritz (2010). The classification results from WEKA are merged with the original corpus and re-imported to ANNIS2.

Qualitative and quantitative error analysis using ANNIS2 In ANNIS2, the classification results can be queried and viewed in context. The anaphoricity predicted by our WEKA classifier, for example, can be confirmed by comparing coreference annotations with classification results as in query (2):

```
(2) anaph_pred="NULL" &
    node & #2 _=_ #1 &
    node & #2 ->IDENT_relation #3
```

Query (2) retrieves every NP #1 that is *incorrectly* classified as non-anaphoric, because it is coextensional with a node #2 that takes #3 as an anaphoric antecedent.

Using the ANNIS visualizations, the errors of the classifier can be manually inspected together with other annotations, and hypotheses regarding contextual features can be developed on this basis. Based on such qualitative error analyses, quantitative studies can be performed, e.g., by calculating contingency tables for novel feature combinations using the match count for different specialized ANNIS2 queries. Significant features can then be integrated into the classification. The classification results can then be re-imported again, and the next cycle begins.

Using 7 contextual features,⁵ the classifier described by Ritz (2010) distinguishes anaphoric from non-anaphoric NPs with 92.66% accuracy. As compared to the baseline (classify all NPs as being non-anaphoric, accuracy 86.37%), this result represents a significant improvement (McNemar test, $p < .01$).

Qualitative analysis (manual inspection of the misclassifications) indicates that grammatical functions (subject, etc.) correlates with anaphoricity. Such hypotheses receive initial support by means of statistical tests that operate on contingency tables obtained from match counts, i.e., a quantitative analysis.

Querying for edge labels allows us to export grammatical functions in the ARFF table, and adding this feature to the classifier indeed increases its performance significantly (McNemar test, $p < .01$, accuracy 92.98%).

⁵NP length, noun form (e.g., proper noun, common noun, or pronoun), named entity type (e.g., person, organization, etc.), phrase form (definite, indefinite, pronominal, etc.), previous mentions, previous mentions of the head noun, the size of the left context, whether the NP is embedded in another NP or not.

3 Conclusion and Outlook

We have presented a cycle of feature extraction and classifier refinement; our workflow combines a user-friendly graphical interface for corpus querying and annotation visualization (ANNIS2), classifier training (WEKA), format conversion (converter site) and the integration of novel corpora to ANNIS2 (Kickstarter).

An important aspect is that this workflow allows for distributed processing by people with different specialization. Kickstarter and ANNIS2 backend, for example, may run on a sever maintained by a technician with little or no NLP background. A linguist or NLP engineer, however, who is granted access to the data via ANNIS2, can develop highly sophisticated classifiers on his local WEKA installation and, using the converter site, he can convert, merge and reintegrate the data with little effort (and without bothering about details of the conversion process).

The workflow described above is not unprecedented (see, e.g., Kermes and Evert, 2001), but it differs from earlier accounts in that it uses a corpus tool that supports multi-layer annotations, and thus allows to combine not only different flat annotations with each other, but also, multiple hierarchical annotations (e.g., phrase structure annotations besides semantic annotations and discourse structural annotations *on the same stretch of primary data*), or annotations originating from different annotation tools (e.g., coreference annotation besides syntax annotation as in our example). The application of ANNIS2 thus introduces a new quality in the number of parameters and contextual dependencies to be processed.

References

- Christian Chiarcos, Stefanie Dipper, Michael Götze, Ulf Leser, Anke Lüdeling, Julia Ritz, and Manfred Stede. 2009. A Flexible Framework for Integrating Annotations from Different Tools and Tagsets. *TAL (Traitement automatique des langues)*, 49(2).
- Christian Chiarcos, Kerstin Eckart, and Julia Ritz. 2010. Creating and exploiting a resource of parallel parses. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 166–171, Uppsala, Sweden, July.
- Stefanie Dipper and Michael Götze. 2005. Accessing heterogeneous linguistic data – Generic XML-based representation and flexible visualization. In *Proceedings of the 2nd Language & Technology Conference 2005*, pages 23–30, Poznan, Poland, April.
- Stefanie Dipper. 2005. XML-based stand-off representation and exploitation of multi-level linguistic annotation. In *Proceedings of Berliner XML Tage 2005 (BXML 2005)*, pages 39–50, Berlin, Germany.
- Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 57–60, New York City, USA.
- Nancy Ide and Keith Suderman. 2007. GrAF: A graph-based format for linguistic annotations. In *Proceedings of The Linguistic Annotation Workshop (LAW) 2007*, pages 1–8, Prague, Czech Republic.
- Hannah Kermes and Stefan Evert. 2001. Exploiting large corpora: A circular process of partial syntactic analysis, corpus query and extraction of lexicographic information. In *Proceedings of the Corpus Linguistics 2001*, pages 332 – 340, Lancaster, UK.
- Miles Osborne and Jason Baldridge. 2004. Ensemble-based active learning for parse selection. In *Proceedings of HLT-NAACL*, pages 89–96.
- Petr Pajas and Jan Štěpánek. 2009. System for querying syntactically annotated corpora. In *Proceedings of ACL-IJCNLP 2009*, pages 33–36, Singapore, August.
- Julia Ritz. 2010. Using tf-idf-related measures for determining the anaphoricity of noun phrases. In *Proceedings of KONVENS 2010*, Saarbrücken, September.
- Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *Proceedings of the 16th International Conference on Machine Learning ICML-99*, pages 406–414.
- Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Linnea Micciulla, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Olga Babko-Malaya, Eduard Hovy, Robert Belvin, and Ann Houston. 2007. OntoNotes Release 1.0. Technical report, Linguistic Data Consortium, Philadelphia.
- Ian H. Witten and Eibe Frank. 2005. *Data mining: Practical machine learning tools and techniques*. Morgan Kaufman, San Francisco, 2nd edition.
- John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR, August.

POS-Tagging of Historical Language Data: First Experiments

Stefanie Dipper
 Institute of Linguistics
 Ruhr-University Bochum
 Germany
 dipper@linguistics.rub.de

Abstract

This paper deals with part-of-speech tagging applied to manuscripts written in Middle High German. We present the results of a set of experiments that involve different levels of token normalization and dialect-specific subcorpora. As expected, tagging with “normalized”, quasi-standardized tokens performs best (accuracy > 91%). Training on slightly simplified word forms or on larger corpora of heterogeneous texts does not result in considerable improvement.

1 Introduction¹

This paper deals with automatic analysis of historical language data, namely part-of-speech (POS) tagging of texts from Middle High German (1050–1350). Analysis of historical languages differs from that of modern languages in two important points.

First, there are no agreed-upon, standardized writing conventions. Instead, characters and symbols used by the writer of some manuscript in parts reflect impacts as different as spatial constraints (parchment is expensive, hence, use of abbreviations seems favorable) or dialect influences (the dialect spoken by the author of the text, or the writer’s dialect, who writes up or copies the text, or even the dialect spoken by the expected readership). This often leads to inconsistent spellings, even within one text written up by one writer.

Second, resources of historical languages are scarce and often not very voluminous, and manuscripts are frequently incomplete or damaged.

¹We would like to thank the anonymous reviewers for helpful comments. The research reported here was supported by Deutsche Forschungsgemeinschaft (DFG), Grant DI 1558/1-1.

These features—data variance and lack of large resources—challenge many analysis tools, whose quality usually depend on the availability of large training samples. “Modern” POS taggers have been used mainly for the annotation of English historical corpora. The “Penn-Helsinki Parsed Corpora of Historical English” (Kroch and Taylor, 2000; Kroch et al., 2004) have been annotated in a bootstrapping approach, which involves successive cycles of manual annotation, training, automatic tagging, followed by manual corrections, etc. The project “GermanC”² uses a state-of-the-art tagger whose lexicon has been filled with historical form variants. In contrast, Rayson et al. (2007) and Pilz et al. (2006) automatically map historical word forms to the corresponding modern word forms, and analyze these by state-of-the-art taggers. The mappings make use of the Soundex algorithm, Edit Distance, or heuristic rules. Rayson et al. (2007) apply this technique for POS tagging, Pilz et al. (2006) for a search engine for texts without standardized spelling.

This paper reports on preliminary experiments in applying a state-of-the-art POS tagger (the TreeTagger, Schmid (1994)) to a corpus of texts from Middle High German (MHG). Our approach is similar to the one by Kroch et al. in that we train and apply the tagger to historical rather than modern word forms. Our tagging experiments make use of a balanced MHG corpus that is created and annotated in the context of the projects “Mittelhochdeutsche Grammatik” and “Referenzkorpus Mittelhochdeutsch”.³ The corpus has been semi-

²<http://www.llc.manchester.ac.uk/research/projects/germanc/>

³<http://www.mittelhochdeutsche-grammatik.de>, <http://www.linguistics.rub.de/mhd/>

	ich dir gelobe. dar zu ne helbe ich dir											
UNICODE	ich	dir	gelobe	.	dar	z ^ö	ne	helbē	ich	dir		
STRICT	ich	dir	gelobe	.	dar	zu\o	ne	helbe\	-ich	dir		
SIMPLE	ich	dir	gelobe	.	dar	zuo	ne	helben	ich	dir		
NORM	ich	dir	gelobe	.	dar	zuo	ne	hilfen	ich	dir		
LEMMA	ich	dû	ge-loben		dâr	zuo	ne	hëlfen	ich	dû		
STTS	PPER	PPER	VVFIN	\$.	ADV	ADV	PTK-	VVFIN	PPER	PPER	NEG	

Figure 1: A small excerpt from *Tristrant* (Magdeburg fragment), along with different types of transcriptions and annotations (screenshot from http://www.hs-augsburg.de/~harsch/germanica/Chronologie/12Jh/Eilhart/eil_tmma.html)

automatically annotated with POS tags, morphology, lemma, and a normalized word form, which represents a virtual historical standard form. The corpus is not annotated with modern word forms.

In this paper, we present the results of a set of experiments that involve different types of tokens (simplified and normalized versions) and dialect-specific subcorpora. Sec. 2 presents detailed information about the corpus and its annotations, Sec. 3 addresses the tagging experiments and results.

2 The Corpus

The corpus is a collection of texts from the 12th–14th centuries, including religious as well as profane texts, prose and verse. The texts have been selected in a way as to cover the period of MHG as optimal as possible. The texts distribute in time, i.e. over the relevant centuries, and in space, coming from a variety of Upper German (UG) and Middle German (MG) dialects. UG dialects were (and are still) spoken in the Southern part of Germany, Switzerland, and Austria. Examples are Bavarian, Swabian, or Alemannic. MG dialects were spoken in the middle part of Germany, e.g. Franconian or Thuringian.

The texts are *diplomatic transcriptions*, i.e., they aim at reproducing a large range of features of the original manuscript or print, such as large initials, or variant letter forms (e.g. short vs. long s: <s> vs. <f>), or abbreviations (e.g. the superscribed hook <‘> can abbreviate -er: *cleid*[‘] stands for *cleider* ‘clothes’⁴).

The original corpus provides two different versions of “word forms”: the diplomatic transcription and a normalized form. For the tagging ex-

periments, we created simplified versions of these forms: “strict” and “simple” transcriptions, and a normalized ASCII-form. In the following, we describe both the original and simplified versions of the word forms. Figure 1 presents an example fragment encoded in the different versions.

- The *Unicode* version is the diplomatic transcription, as produced by human transcribers. It renders the manuscript very closely and, e.g., distinguishes short vs. long s, abbreviations, etc.
- The *strict* version is a slightly modified version of the original transcription which uses ASCII characters only. Instead of letters with diacritics or superposed characters (ö, ü), it uses ASCII characters combined with the backslash as an escape character (o\”, u\o). Ligatures (æ) are marked by an underscore (a_e), & is mapped to e_t, þ to t_h.
- The *simple* version abstracts away from many of the diplomatic and/or dialectal idiosyncrasies. Characters are mapped to lower case, all kinds of accents or other diacritics are removed. Character combinations are mapped to the base characters (ü, æ become uo, ae, respectively). Abbreviations are spelt out (e.g., the ‘-hook becomes er).
- Finally, the *norm(alized)* version is an artificial standard form, similar to the citation forms used in lexicons of MHG, such as Lexer (1872). The normalized form abstracts away completely from dialectal sound/grapheme variance. It has been semi-automatically generated by a tool developed by the project “Mittelhochdeutsche Grammatik”. The tool exploits lemma and morphological information in combination with symbolic rules that encode linguistic knowledge about historical dialects (Klein, 2001). We use a simplified ASCII version of the normalized form, with modifications similar to the ones of the simple transcription version.

⁴<‘> can also stand for re, r, and rarely for ri, ir. We replace it unambiguously by er, which seems to be the most frequent case.

Texts	Tokens	Types		
		strict	simple	norm
51 total	211,000	40,500 .19	34,500 .16	20,500 .10
27 MG	91,000	22,000 .24	19,000 .21	13,000 .14
20 UG	67,000	15,000 .22	13,500 .20	8,500 .13
4 mixed	53,000			

Table 1: Number of tokens and types in the Middle High German corpus. Below each type figure, the type-token ratio is given.

Table 1 displays some statistics of the current state of the corpus. The first column shows that there are currently 51 texts in total, with a total of around 211,000 tokens. The shortest text contains only 51 tokens, the longest one 25,000 tokens. 27 texts are from MG dialects and 20 from UG dialects. 4 texts are classified as “mixed”, because they show mixed dialectal features, or are composed of fragments of different dialects.

As Table 1 shows, the numbers of types are somewhat reduced if strict (diplomatic) word forms are mapped to simple forms. Comparing strict and normalized types, the numbers are roughly cut in half. This benefits current taggers, as it reduces the problem of data sparseness to some extent. The question is, however, how reliably the normalized form can be generated automatically. The current tool requires a considerable amount of manual intervention during the analyses of lemma and morphology.

MG texts seem more diverse than UG texts: Despite the fact that the MG subcorpus is larger than the UG subcorpus, it has a higher type/token ratio.

The texts are further annotated with POS tags. The original POS tagset comprises more than 100 tags and encodes very fine-grained information. For instance, there are 17 different tags for verbs, whose main purpose is to indicate the inflection class that the verb belongs to. For the experiments described in this paper, these POS tags were mapped automatically to a modified version of the STTS tagset, the de-facto standard tagset for modern German corpora (Schiller et al. (1999); see Fig. 1).⁵

⁵The STTS modifications are:

(i) An underspecified tag for the demonstrative or relative pronoun *der* has been introduced. The distinction between both types of pronouns can be made rather easily for modern German texts: relative pronouns induce subordinate word order, whereas demonstrative pronouns do not. In MHG, the

3 The Experiments

For the experiments, we performed a 10-fold cross-validation. The split was done in blocks of 10 sentences (or “units” of a fixed number of words, if no punctuation marks were available⁶). Within each block, one sentence was randomly extracted and held out for the evaluation.

For the analysis, we used the TreeTagger, since it takes suffix information into account. Thanks to this property, the TreeTagger can profit from units smaller than words, which seems favorable for data with high variance in spelling.

In our experiments, we modified two parameters during training: (i) word forms: *strict*, *simple*, *norm*; (ii) dialects: *all*, *MG*, *UG* (i.e., training data consists of the entire corpus, or the MG subcorpus, or the UG subcorpus). For instance, in one setting the tagger is trained on the strict forms of MG data.

For the evaluation, we introduced a further parameter: (iii) tagger: *specific*, *general*, *incorrect*. In the specific setting, the trained tagger is applied to the “correct”, specific data (e.g., the tagger trained on strict-MG data is evaluated on strict-MG data). In the general setting, the tagger trained on the entire corpus is applied to some subcorpus. Finally, in the “incorrect” setting, the tagger trained on MG data is evaluated on UG data, and vice versa.

The first evaluation setting is straightforward. Setting two gives an impression of which performance we can expect if we apply a tagger that has been trained on a larger data set, which, however, consists of heterogeneous dialect texts. Setting three shows the extent to which performance can degrade in a kind of worst case scenario. In addition, settings three allows us to compare the

position of the verb, which marks subordinate word order, was not as fixed as nowadays. Hence, this property should not be used as a criterion.

(ii) General tags PW, PI, and KO are used rather than PWS/PWAT, or PIS/PIAT, or KON/KOUS etc., because the original tagset does not allow to reconstruct the distinction.

(iii) All adjectives are tagged with the underspecified tag ADJ. Predicative adjectives can be inflected in MHG and, thus, a mapping to ADJA/ADJD is not easily definable.

(iv) Finally, the suffix _LAT subtype was introduced to mark Latin words and their POS tags (e.g. V_LAT for Latin verbs). These occur quite frequently in historical texts. Our corpus contains a total of 5,500 Latin words (= 2.6% of all tokens). In the MG texts, 5.3% of the tokens are Latin, whereas in the UG texts, only 0.9% are Latin.

⁶Punctuation marks in historical texts do not necessarily mark sentence or phrase boundaries. Nevertheless, they probably can serve as indicators of unit boundaries at least as well as randomly-picked boundary positions.

Dialect	Tagger	Word Forms		
		strict	simple	norm
MG	<i>specific</i>	86.62 ± 0.63	87.65 ± 0.59	91.43 ± 0.39
	<i>general</i>	86.92 ± 0.64	87.69 ± 0.58	91.66 ± 0.47
	<i>incorrect</i>	65.48 ± 0.73	71.20 ± 0.56	81.59 ± 0.44
	<i>unknowns</i>	59.71 ± 1.84	62.26 ± 2.18	68.14 ± 1.34
UG	<i>specific</i>	89.16 ± 0.75	89.58 ± 0.72	92.91 ± 0.29
	<i>general</i>	88.88 ± 0.68	89.45 ± 0.59	92.83 ± 0.39
	<i>incorrect</i>	77.81 ± 0.80	79.76 ± 0.57	89.43 ± 0.49
	<i>unknowns</i>	62.77 ± 1.77	64.81 ± 2.62	70.46 ± 2.21

Table 2: Results of 18 test runs, based on different types of word forms, dialect subcorpora, and taggers, and 6 evaluations of unknown tokens. For each scenario, mean and standard deviation of per-word accuracy across the 10 folds are given (all values are percentages).

impact of the normalization step: Since normalization is supposed to level out dialectal differences, we expect less deterioration of performance with norm forms than with strict or simple forms.

The results of the different scenarios are summarized in Table 2. For each scenario, mean and standard deviation of per-word accuracy across the 10 folds are given.

The table shows that taggers achieve higher scores with UG data than MG data, in all scenarios. This result is somewhat unexpected since the size of the UG subcorpus is only 75% of that of the MG subcorpus. However, as we have seen, MG data is more diverse and has a higher type/token ratio.

With respect to the different types of word forms, tagging with normalized forms turns out best, as expected. The differences between strict and simple forms are surprisingly small, given the fact that the size of the “simpler vocabulary” is only around 85% of the “strict vocabulary”.⁷ The wide difference between simple and normalized forms reflects the fact that the level of standardization as introduced by the normalization step concerns not just minor features such as accents or ligatures but also inflectional endings and sound changes.

Comparing the three types of taggers, the table clearly shows that the specific and general taggers perform considerably better than the incorrect ones. As expected, the differences between the taggers are less pronounced with normalized word forms. Interestingly, the specific and general

tagger variants do *not* differ significantly in most of the scenarios, despite the fact that the general taggers have been trained on a larger data set.⁸

Finally we evaluated the performance of the specific taggers on unknown words. The results show that performance degrades considerably, by 22.5 (with the UG-norm tagger) up to 26.9 percentage points (with the MG-strict tagger).

4 Summary and Outlook

We presented a set of experiments in POS tagging of historical data. The aim of this enterprise is to evaluate how well a state-of-the-art tagger, such as the TreeTagger, performs in different kinds of scenarios. The results cannot directly compared to results from modern German, though: Our corpora are rather small; historical data is considerably more diverse than modern data; and we used a modified version of the STTS.

As future steps, we will perform a detailed error analysis: Which tags are especially hard to learn, which tags are difficult to distinguish? Can certain errors be traced back to dialectal properties of the language? Is there an impact of time of origin of a manuscript?

To reduce variance of the data, without requiring complete normalization of all words, we plan to investigate a hybrid approach, by evaluating whether it is helpful to normalize function words only and keep content words unmodified. Since function words are closed classes, it might be possible to successfully normalize these words automatically, without manual intervention.

⁷ Maybe this outcome can be attributed to the TreeTagger, which possibly performs similar simplifications internally. All word form results differ significantly from each other, though.

⁸ The general taggers perform significantly better than the corresponding specific taggers when they are evaluated on MG-norm and MG-strict data (paired t-test; MG-norm data: $t=4.48$, $df=9$, $p<.01$; MG-strict data: $t=4.20$, $df=9$, $p<.01$).

References

- Thomas Klein. 2001. Vom lemmatisierten Index zur Grammatik. In Stephan Moser, Peter Stahl, Werner Wegstein, and Norbert Richard Wolf, editors, *Maschinelle Verarbeitung altdeutscher Texte V. Beiträge zum Fünften Internationalen Symposium Würzburg 4.-6. März 1997*, pages 83–103. Tübingen: Niemeyer.
- Anthony Kroch and Ann Taylor. 2000. Penn-Helsinki parsed corpus of Middle English. Second edition, <http://www.ling.upenn.edu/hist-corpora/PPCME2-RELEASE-2/>.
- Anthony Kroch, Beatrice Santorini, and Lauren Delfs. 2004. Penn-Helsinki parsed corpus of Early Modern English. <http://www.ling.upenn.edu/hist-corpora/PPCEME-RELEASE-1/>.
- Matthias Lexer. 1872. *Mittelhochdeutsches Handwörterbuch*. Leipzig. 3 Volumes 1872–1878. Reprint: Hirzel, Stuttgart 1992.
- Thomas Pilz, Wolfram Luther, Ulrich Ammon, and Norbert Fuhr. 2006. Rule-based search in text databases with nonstandard orthography. *Literary and Linguistic Computing*, 21:179–86.
- Paul Rayson, Dawn Archer, Alistair Baron, Jonathan Culpeper, and Nicholas Smith. 2007. Tagging the Bard: Evaluating the accuracy of a modern POS tagger on Early Modern English corpora. In *Proceedings of Corpus Linguistics 2007*, University of Birmingham, UK.
- Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset). Technical report, University of Stuttgart and University of Tübingen.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*.

Trainable Tree Distance and an application to Question Categorisation*

Martin Emms

School of Computer Science and Statistics

Trinity College

Dublin, Ireland

Martin.Emms@tcd.ie

Abstract

Continuing a line of work initiated in Boyer et al. (2007), the generalisation of stochastic string distance to a stochastic tree distance, specifically to stochastic Tai distance, is considered. An issue in modifying the Zhang/Shasha tree-distance algorithm to the stochastic variants is noted, a Viterbi EM cost-adaptation algorithm for this distance is proposed and a counter-example noted to an all-paths EM proposal. Experiments are reported in which a k-NN categorisation algorithm is applied to a semantically categorised, syntactically annotated corpus. We show that a 67.7% base-line using standard unit-costs can be improved to 72.5% by cost adaptation.

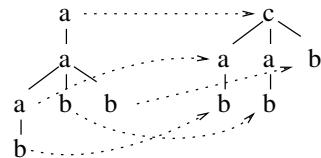
1 Theory and Algorithms

The classification of syntactic structures into semantic categories arises in a number of settings. A possible approach to such a classifier is to compute a category for a test item based on its distances to a set of k nearest neighbours in a pre-categorised example set. This paper takes such an approach, deploying variants of *tree-distance*, a measure which has been used with some success in tasks such as Question-Answering, Entailment Recognition and Semantic Role Labelling (Punyakanok et al., 2004; Kouylekov and Magnini, 2005; Emms, 2006a; Emms, 2006b; Franco-Peña, 2010). An issue which will be considered is how to *adapt* the atomic costs underlying the tree-distance measure.

Tai (1979) first proposed a tree-distance measure. Where S and T are ordered, labelled trees, a *Tai* mapping is a *partial, 1-to-1* mapping σ from

* This work was supported in part through the *Centre for Next Generation Localisation* by the Science Foundation Ireland

the nodes of S to the nodes of T , which respects *left-to-right order* and *ancestry*¹, such as



A cost can be assigned to a mapping σ based on the nodes of S and T which are not 'touched' by σ , and the set of pairs (i, j) in σ . The *Tai- or tree-distance* $\Delta(S, T)$ is defined as the cost of the least-costly Tai mapping between S and T . Equivalently, *tree-edit* operations may be specified, and the distance defined by the cost of the least costly sequence of edit operations transforming S into T , compactly recorded as an edit-script:

operation	edit-script element
$m'(\vec{l}, \mathbf{m}(\vec{d}), \vec{r}) \rightarrow m'(\vec{l}, \vec{d}, \vec{r})$	(m, λ)
$m'(\vec{l}, \vec{d}, \vec{r}) \rightarrow m'(\vec{l}, \mathbf{m}(\vec{d}), \vec{r})$	(λ, m)
$\mathbf{m}(\vec{d}) \rightarrow \mathbf{m}'(\vec{d})$	(m, m')

An edit-script can be seen as a serialization of a mapping, and the distances via scripts and via mappings are equivalent (Zhang and Shasha, 1989).

If strings are treated as vertical trees, the Tai distance becomes the standard string distance (Wagner and Fischer, 1974). Ristad and Yianilos (1998) pioneered a probabilistic perspective on string distance via a model in which there is a probability distribution p on edit-script components, and $P(e_1 \dots e_n) = \prod_i p(e_i)$. It is natural to consider how this probabilistic perspective can be applied to tree-distance, and the simplest possibility is to use exactly the same model of edit-script probability, leading to²:

¹so if (i_1, j_1) and (i_2, j_2) are in the mapping, then (T1) $left(i_1, i_2)$ iff $left(j_1, j_2)$ and (T2) $anc(i_1, i_2)$ iff $anc(j_1, j_2)$

² Δ^A was proposed by Boyer et al. (2007)

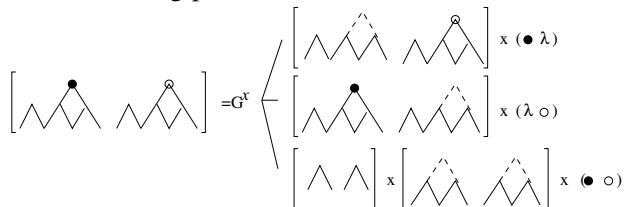
Definition 1.1 (All-paths and Viterbi stochastic Tai distance) $\Delta^A(S, T)$ is the sum of the probabilities of all edit-scripts which represent a Tai-mapping from S to T ; $\Delta^V(S, T)$ is the probability of the most probable edit-script

Computing Δ^A and Δ^V We have adapted the Zhang/Shasha algorithm for Tai-distance to the stochastic case. The algorithm operates on the left-to-right post-order traversals of trees³. If i is (the index of) a node of the tree, let $\gamma(i)$ be its label, i_l be the leaf reached by following the left-branch down, and $S[i]$ be the sub-tree of S rooted at i . If i' is a member of $S[i]$, the prefix $i_l..i'$ of the traversal of $S[i]$ can be seen as a *forest* of subtrees. Considering the mappings between such forests, a case distinction can made on the possible final element of any script serializing the mapping, giving the following decomposition for the calculation of Δ^V and Δ^A

Lemma 1.1 where G^V is the max operation, and G^A is the sum operation, for $x \in \{V, A\}$

$$\Delta^x(i_l..i', j_l..j') = G^x \begin{cases} \Delta^x(i_l..i' - 1, j_l..j') \times p(\gamma(i'), \lambda) \\ \Delta^x(i_l..i', j_l..j' - 1) \times p(\lambda, \gamma(j')) \\ \Delta^x(i_l..i'_l - 1, j_l..j'_l - 1) \times \\ \underbrace{\Delta^x(i'_l..i' - 1, j'_l..j' - 1) \times p(\gamma(i'), \gamma(j'))}_{\Delta_M^x(i'_l..i', j'_l..j')} \end{cases}$$

The following picture illustrates this



For any leaf i , the highest node k such that $i = k_l$ is a *key-root*, and $KR(S)$ is the key-roots of S ordered by post-order traversal. For $x \in \{A, V\}$, the main loop of TD^x is then essentially

for $i \in KR(S)$, $j \in KR(T)$
for $i' : i_l \leq i' \leq i$, $j' : j_l \leq j' \leq j$,
compute $\Delta^x(i_l..i', j_l..j')$ via Lemma 1.1

computing a series of forest distance tables, whilst reading and updating a persistent tree table. Space precludes further details except to note the subtlety in TD^A that to avoid double counting, the tree table must store values only for mappings between trees with matched or substituted roots (the

³so parent follows children

$\Delta_M^A(i_l..i', j_l..j')$ term in Lemma 1.1), unlike the Zhang/Shasha algorithm, where it stores the true tree-distance⁴.

TD^A and TD^V work under a negated logarithmic mapping⁵, with $\times/\max/\sum$ mapped to $+/min/\logsum$ ⁶. Where Σ is the label alphabet, a *cost table* C of dimensions $(|\Sigma|+1) \times (|\Sigma|+1)$ represents (neg-logs of) atomic edit operation, with first column and row for deletions and insertions. For Δ^V and Δ^A , the probabilities represented in C should sum to 1. For TD^V , the neg-log mapping is never inverted and TD^V can be run with arbitrary C and calculates then the standard non-stochastic Tai distance. The *unit-cost* table, C_{01} , has 0 on the diagonal and 1 everywhere else.

Adapting costs We are interested in putting tree-distance measures to work in deriving a category for an uncategorised item, using an *example-set* of categorised examples, via the k nearest-neighbour (kNN) algorithm. The performance of the kNN classification algorithm will vary with cost-table C and Expectation-Maximisation (EM) is a possible approach to setting C . Given a corpus of training pairs, let the *brute-force all-paths EM algorithm*, EM_{bf}^A , consists in iterations of: (E) generate a virtual corpus of scripts by treating each training pair (S, T) as standing for the edit-scripts \mathcal{A} , which can relate S to T , weighting each by its conditional probability $P(\mathcal{A})/\Delta^A(S, T)$, under current costs C and (M) apply maximum likelihood estimation to the virtual corpus to derive a new cost-table. EM_{bf}^A is not feasible. Let EM^V be a Viterbi variant of this working with a virtual corpus of *best*-scripts only, effectively weighting each by the proportion it represents of the all-paths sum, $\Delta^V(S, T)/\Delta^A(S, T)$. Space precludes further details of EM^V . Such Viterbi training variants have been found beneficial, for example in the context of parameter training for PCFGs (Benedí and Sánchez, 2005). The training set for EM^V is tree pairs (S, T) , where for each *example-set* tree S, T is a nearest same-category neighbour. EM^V increases the edit-script probability for scripts linking these trees, lessening their distance. Note that without the stochastic constraints on C , the dis-

⁴Boyer et al. (2007) present somewhat unclear algorithms for Δ^A , not explicitly as extensions of the Zhang/Shasha algorithm, and do not remark this double-counting subtlety. Their on-line implementation (SEDiL, 2008) can compute incorrect values and this work uses our own implementation of the algorithms here outlined.

⁵ $x = \text{neg-log}(p)$ iff $p = 2^{-x}$

⁶ $\logsum(x_1 \dots x_n) = -\log(\sum_i(2^{-x_i}))$

tance via TD^V could be minimised to zero by setting all costs to zero, but this would be of no value in improving the categorisation performance.

To initialize EM^V , let $\mathcal{C}_u(d)$ stand for a stochastically valid cost-table, with the additional properties that (i) all diagonal entries are equal (ii) all non-diagonal entries are equal (iii) diagonal entries are d times more probable than non-diagonal. As a *smoothing* option concerning a table \mathcal{C} derived by EM^V , let \mathcal{C}_λ be its interpolation with the original $\mathcal{C}_u(d)$ as follows

$$2^{-\mathcal{C}_\lambda[x][y]} = \lambda(2^{-\mathcal{C}[x][y]}) + (1 - \lambda)(2^{-\mathcal{C}_u(d)[x][y]})$$

For stochastic string-distance Ristad and Yianilos (1998) provided a feasible equivalent to EM_{bf}^A : for each training pair (s, t) , first *position-dependent* expectations $\mathcal{E}[i][j](x, y)$ are computed, then later summed into position-independent expectations. Boyer et al. (2007) contains a proposal in a similar spirit to provide a feasible equivalent to EM_{bf}^A but the proposal factorizes the problem in a way which is invalid given the ancestry-preservation aspect of Tai mappings⁷. For example, using a post-fix notation subscripting by post-order position, let $t_1 = (\cdot_1 (\cdot_2 \cdot_3 m_4) \cdot_5 \cdot_6)$, $t_2 = ((\cdot_1 \cdot_2) (\cdot_3 m'_4) (\cdot_5 \cdot_6) \cdot_7)$ (from fig 3 of their paper). They propose to calculate a swap expectation $\mathcal{E}[4, 4](m, m')$ by

$$[\Delta^A((\cdot_1), (\cdot_1 \cdot_2)) \times [\Delta^A((\cdot_2)(\cdot_3), (\cdot_3)) \times p(m, m')] \times \Delta^A((\cdot_5 \cdot_6), ((\cdot_5 \cdot_6) \cdot_7))] / \Delta^A(t_1, t_2)$$

But $\Delta^A((\cdot_5 \cdot_6), ((\cdot_5 \cdot_6) \cdot_7))$ will contain contributions from scripts which map t_1 's \cdot_6 , an ancestor of m_4 , to t_2 's \cdot_6 , a non-ancestor of m'_4 , and these should not contribute to $\mathcal{E}[4, 4](m, m')$.

2 Experiments

QuestionBank (QB) is a hand-corrected treebank for questions (Judge, 2006). A substantial percentage of the questions in QB are taken from a corpus of semantically categorised, syntactically unannotated questions (CCG, 2001). From these two corpora we created a corpus of 2755 semantically categorised, syntactically analysed questions⁸, spread over the semantic categories as follows⁹: HUM(23.5%), ENTY(22.5%), DESC(19.4%), NUM(16.7%), LOC(16.5%) and ABBR(1.4%).

⁷A fact which they concede p.c.

⁸available at www.scss.tcd.ie/Martin.Emms/quest_cat

⁹See (CCG, 2001) for details of the semantic category labels

This corpus was used in a number of experiments on kNN classification using the tree-distance TD^V algorithm, with various cost tables. In each case 10-fold cross-validation was used with a 9:1 example-set/test-set split.

Figure 1 shows some results of a first set of experiments, with unit-costs and then with some stochastic variants. For the stochastic variants, the cost initialisation was $\mathcal{C}_u(3)$ in each case.

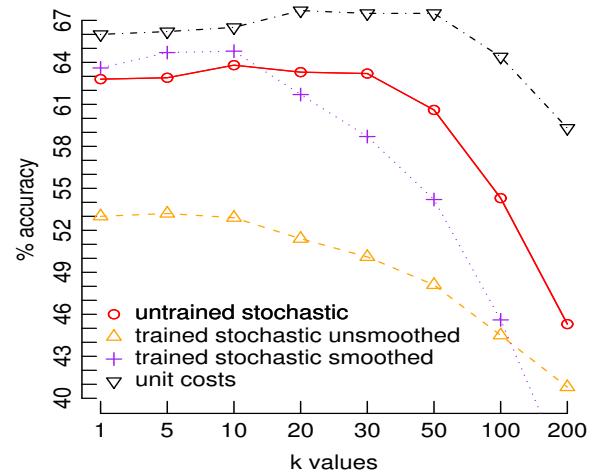


Figure 1: *Categorisation performance with unit costs and some stochastic variants*

The first thing to note is that performance with unit-costs (∇ , max. 67.7%) exceeds performance with the non-adapted $\mathcal{C}_u(3)$ costs (\circ , max. 63.8%). Though not shown, this remains the case with far higher settings of the diagonal factor. Performance after applying EM^V to adapt costs (\triangle , max. 53.2%) is worse than the initial performance (\circ , max. 63.8%). A Leave-One-Out evaluation, in which *example-set* items are categorised using the method on the remainder of the *example-set*, gives accuracies of 91% to 99%, indicating EM^V has made the best-scripts connecting the training pairs *too* probable, *over-fitting* the cost table. The vocabulary is sufficiently thinly spread over the training pairs that it's quite easy for the learning algorithm to fix costs which make almost everything but exactly the training pairs have zero probability. The performance when smoothing is applied ($+$, max. 64.8%), interpolating the adapted costs with the initial cost, with $\lambda = 0.99$, is considerably higher than without smoothing (\triangle), attains a slightly higher maximum than with unadapted costs (\circ), but is still worse than with unit costs (∇).

The following is a selection from the top 1% of adapted swap costs.

8.50	?	.	12.31	The	the
8.93	NNP	NN	12.65	you	I
9.47	VBD	VBZ	13.60	can	do
9.51	NNS	NN	13.83	many	much
9.78	a	the	13.92	city	state
11.03	was	is	13.93	city	country
11.03	's	is			

These learned preferences are to some extent intuitive, exchanging punctuation marks, words differing only by capitalisation, related parts of speech, verbs and their contractions and so on. One might expect this discounting of these swaps relative to others to assist the categorisation, though the results reported so far indicate that it did not. A stochastically valid cost table cannot have zero costs on the diagonal, and even with a very high ratio between the diagonal and off-diagonal probabilities, the diagonal costs are not negligible. Perhaps this mitigates against success and invites consideration of outcomes if a final step is applied in which all the entries on the diagonal are zeroed. In work on adapting cost-tables for a stochastic version of *string distance* used in duplicate detection, Bilenko and Mooney (2003) used essentially this same approach. Figure 2 shows outcomes when the trained and smoothed costs finally have the diagonal zeroed.

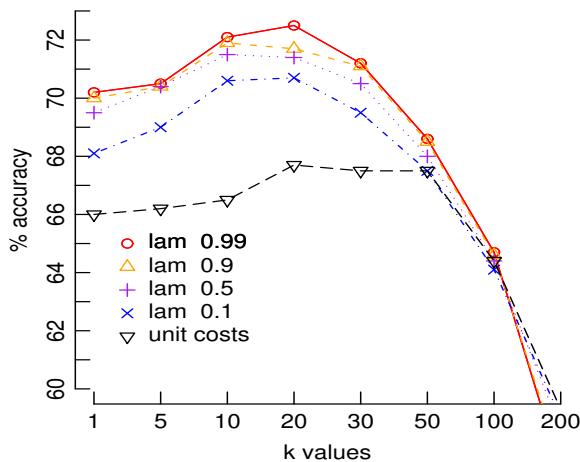


Figure 2: Categorisation performance: adapted costs with smoothing and zeroing

The (∇) series once again shows the outcomes with unit-costs whilst the other series show outcomes obtained with costs adapted by EM^V , smoothed at various levels of interpolation ($\lambda \in \{0.99, 0.9, 0.5, 0.1\}$) and with the diagonal ze-

roed. Now the unit costs base-line is clearly outperformed, the best result being 72.5% ($k = 20$, $\lambda = 0.99$), as compared to 67.5% for unit-costs ($k = 20$)

3 Comparisons and Conclusions

Collins and Duffy (2001) proposed the $SST(S, T)$ tree-kernel 'similarity': a product in an infinite vector space, the dimensions of which are counts $c(t)$ of tree substructures t , each $c(t)$ weighted by a decay factor $\gamma^{size(t)}$, $0 < \gamma \leq 1$, and it has been applied to tree classification tasks (Quarternoni et al., 2007). If the negation of $SST(S, T)$ is used as an alternative to $\Delta^V(S, T)$ in the kNN algorithm, we found worse results are obtained¹⁰, 64% – 69.4%, with maximum at $k = 10$. However, deploying $SST(S, T)$ as a kernel in one-vs-one SVM classification¹¹, a considerably higher value, 81.3%, was obtained.

Thus, although we have shown a way to adapt the costs used by the tree-distance measure which improves the kNN classification performance from 67.7% to 72.5%, the performance is less than obtained using tree-kernels and SVM classification. As to the reasons for this difference and whether it is insuperable one can only speculate. The data set was relatively small and it remains for future work to see whether on larger data-sets the outcomes are less dependent on smoothing considerations and whether the kNN accuracy increases. The one-vs-one SVM approach to n -way classification trains $n(n - 1)/2$ binary classifiers, whereas the approach described here has one cost adaptation for all the categories, and a possibility would be to do class-specific cost adaptation, in a fashion similar to Paredes and Vidal (2006).

One topic for future work is to consider how this proposal for cost adaptation relates to other recent proposals concerning adaptive tree measures (Takasu et al., 2007; Dalvi et al., 2009) as well as to consider cost-adaptation outcomes in some of the other areas in which tree-distance has been applied.

¹⁰using the SVMLIGHTTK (2003) implementation

¹¹using the libsvm (2003) implementation, with decay $\gamma = 0.4$, slack $C = 2.0$

References

- José-Miguel Benedí and Joan-Andreu Sánchez. 2005. Estimation of stochastic context-free grammars and their use as language models. *Computer Speech and Language*, 19(3):249–274, July.
- Mikhail Bilenko and Raymond J. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pages 39–48.
- Laurent Boyer, Amaury Habrard, and Marc Sebban. 2007. Learning metrics between tree structured data: Application to image recognition. In *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, pages 54–66.
- CCG. 2001. corpus of classified questions by Cognitive Computation Group, University of Illinois 12r.cs.uiuc.edu/~cogcomp/Data/QA/QC.
- Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS14)*.
- Nilesh Dalvi, Philip Bohannon, and Fei Sha. 2009. Robust web extraction: an approach based on a probabilistic tree-edit model. In *SIGMOD '09: Proceedings of the 35th SIGMOD international conference on Management of data*, pages 335–348, New York, NY, USA. ACM.
- Martin Emms. 2006a. Clustering by tree distance for parse tree normalisation. In *Proceedings of NLUCS 2006*, pages 91–100.
- Martin Emms. 2006b. Variants of tree similarity in a question answering task. In *Proceedings of the Workshop on Linguistic Distances, held in conjunction with COLING 2006*, pages 100–108, Sydney, Australia, July. Association for Computational Linguistics.
- Hector-Hugo Franco-Peña. 2010. Edit tree distance alignments for semantic role labelling. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 79–84, Uppsala, Sweden, July. Association for Computational Linguistics.
- John Judge. 2006. *Adapting and Developing Linguistic Resources for Question Answering*. Ph.D. thesis, Dublin City University.
- Milen Kouylekov and Bernardo Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In Ido Dagan, Oren Glickman, and Bernardo Magnini, editors, *Proceedings of the first Pascal Recognising Textual Entailment Challenge Workshop*.
- libsvm. 2003. library for svm www.csie.ntu.edu.tw/~cjlin/libsvm.
- Roberto Paredes and Enrique Vidal. 2006. Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1100–1110.
- Vasin Punyakanok, Dan Roth, and Wen tau Yih. 2004. Natural language inference via dependency tree mapping: An application to question answering. *Computational Linguistics*.
- Silvia Quarternoni, Alessandro Moschitti, Suresh Manandhar, and Roberto Basili. 2007. Advanced structural representations for question classification and answer re-ranking. In *Advances in Information Retrieval, proceedings of ECIR 2007*. Springer.
- Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(5):522–532, May.
- SEDiL. 2008. Software for computing stochastic tree distance <http://labh-curien.univ-st-etienne.fr/informatique/SEDiL>.
- SVMLIGHTTK. 2003. tree-kernel software disi.unitn.it/moschitti/Tree-Kernel.htm.
- Kuo-Chung Tai. 1979. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):433.
- Atsuhiro Takasu, Daiji Fukagawa, and Tatsuya Akutsu. 2007. Statistical learning algorithm for tree similarity. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 667–672, Washington, DC, USA. IEEE Computer Society.
- Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173, January.
- Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, 18:1245–1262.

Training and Evaluating a German Named Entity Recognizer with Semantic Generalization

Manaal Faruqui

Dept. of Computer Science and Engineering
Indian Institute of Technology
Kharagpur, India
manaal.iitkgp@gmail.com

Sebastian Padó

Maschinelle Sprachverarbeitung
Universität Stuttgart
Stuttgart, Germany
pado@ims.uni-stuttgart.de

Abstract

We present a freely available optimized Named Entity Recognizer (NER) for German. It alleviates the small size of available NER training corpora for German with distributional generalization features trained on large unlabelled corpora. We vary the size and source of the generalization corpus and find improvements of 6% F_1 score (in-domain) and 9% (out-of-domain) over simple supervised training.

1 Introduction

Named Entity Recognition is an important pre-processing step for many NLP tasks. It finds usage in applications like Textual Entailment, Question Answering, and Information Extraction. As is often the case for NLP tasks, most of the work has been done for English. To our knowledge, at this time there is no single “off-the-shelf” NER system for German freely available for academic purposes.

A major reason for this situation is the (un-) availability of labelled development data in the respective languages. For English, many large corpora annotated with named entities are available from a number of shared tasks and bakeoffs, including CoNLL 2003, MUC 2006/2007 and ACE 2008. For German, the only available dataset for NER seems to be the data from the CoNLL 2003 shared task on “Language-Independent Named Entity Recognition” (Tjong Kim Sang and De Meulder, 2003).

The German training part of the CoNLL 2003 data consists only of a total of 220,000 tokens. This is fairly small, but there must be a language-specific aspect at play as well: Even though the amount of training data for English is roughly comparable, the recall of the best system on English data, at 89%, is 25% higher than when trained on German data with 64% (Florian et al., 2003). We hypothesize that this difference is primarily due to the higher morphological complexity of German. Generally,

this puts a higher strain on the lemmatization, and where lemmatization fails, tokens in the test set may simply be unknown. Also, morphological features, which can be learned from comparatively little data, are presumably less predictive for German than they are for English. For example, capitalization is a good predictor of NERs in English, where common nouns are not capitalized. In German, on the other hand, all nouns are capitalized, but most of them are not NEs.

While feature engineering for German is clearly one way out of this situation, the scarcity of labelled data remains a problem since it can lead to overfitting. In this paper, we therefore investigate an alternative strategy, namely *semantic generalization*. We acquire semantic similarities from large, unlabelled corpora that can support the generalization of predictions to new, unseen words in the test set while avoiding overfitting. Our contribution is primarily in evaluation and system building. We train the Stanford NER system (Finkel and Manning, 2009) on different German generalization corpora. We evaluate on both in-domain and out-of-domain data, assessing the impact of generalization corpus size and quality. We make the system with optimal parameters freely available for academic purposes. It is, to our knowledge, among the best available German NERs.

2 Named Entity Recognition with Semantic Generalization

We use Stanford’s Named Entity Recognition system¹ which uses a linear-chain Conditional Random Field to predict the most likely sequence of NE labels (Finkel and Manning, 2009). It uses a variety of features, including the word, lemma, and POS tag of the current word and its context, n -gram features, and “word shape” (capitalization, numbers, etc.).

¹<http://nlp.stanford.edu/software/>

Importantly, the system supports the inclusion of distributional similarity features that are trained on an unrelated large corpus. These features measure how similar a token is to another in terms of its occurrences in the document and can help in classifying previously unseen words, under the assumption that strong semantic similarity corresponds to the same named entity classification. Specifically, the Stanford NER system is designed to work with the clustering scheme proposed by Clark (2003) which combines standard distributional similarity with morphological similarity to cover infrequent words for which distributional information alone is unreliable.² As is generally the case with clustering approaches, the number of clusters is a free parameter. The time complexity of the clustering is linear in the corpus size, but quadratic in the number of clusters.

To illustrate the benefit, imagine that the word “Deutschland” is tagged as location in the training set, and that the test set contains the previously unseen words “Ostdeutschland” and “Westdeutschland”. During clustering, we expect that “Ostdeutschland” and “Westdeutschland” are distributed similarly to “Deutschland”, or are at least morphologically very similar, and will therefore end up in the same cluster. In consequence, these two words will be treated as similar terms to “Deutschland” and therefore also tagged as LOC.

3 Datasets

German corpus with NER annotation. To our knowledge, the only large German corpus with NER annotation was created for the shared task “Language-Independent Named Entity Recognition” at CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003). The German data is a collection of articles from the Frankfurter Rundschau newspaper annotated with four entity types: *person* (PER), *location* (LOC), *organisation* (ORG) & *miscellaneous* (MISC). MISC includes, for example, NE-derived adjectives, events, and nationalities.³ The data is divided into a training set, a development set, and a test set. The training set contains 553 documents and approximately 220,000 tokens. The development set (TestA) and test set (TestB) comprise 155 and 201 documents, respectively, with 55,000 tokens each.

²Clark’s system is available from <http://www.cs.rhul.ac.uk/home/alexc/pos2.tar.gz>

³See <http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt> for annotation guidelines.

Large unlabelled German corpora. For the semantic generalization step, we contrast two corpora that are representative of two widely available classes of corpora. The first corpus, the Huge German Corpus (HGC), consists of approximately 175M tokens of German newspaper text. The HGC is a relatively clean corpus and close in genre to the CoNLL data, which are also newswire. The second corpus is deWac (Baroni et al., 2009), a web-crawled corpus containing about 1.9M documents from 11,000 different domains totalling 1.71B tokens. deWac is very large, but may contain ungrammatical language, and is less similar to the CoNLL data than HGC.

4 Exp. 1: Testing on In-Domain Data

In this experiment, we replicate the CoNLL 2003 setup: We train the NER system on the training set, experiment with different generalization settings while evaluating on the TestA development set, and validate the best models on the TestB test set. We tag and lemmatize both with TreeTagger (Schmid, 1994). We report precision, recall, and F₁ as provided by the CoNLL scorer.

Without any semantic generalization, on TestA we obtain a precision of 80.9%, a recall of 58.8%, and an F-Score of 68.1%. The poor recall corresponds to our expectations for the small size of the training set, and the experiences from CoNLL 2003. It also results in a low overall F₁ score.

For generalization, we apply Clark’s (2003) clustering method to HGC and deWac. For each corpus, we vary two parameters: (a), the amount of generalization data; and (b), the number of clusters created. Following Clark (p.c.), we expect good performance for k clusters when $k^3 \approx n$ where n is the size of the generalization corpus. This leads us to consider at most 600 clusters, and between 10M and 175M tokens, which corresponds to the full size of the HGC and about 10% of deWac.⁴

Table 1 shows the results for using the HGC as generalization corpus. Already the use of 10M tokens for generalization leads to a drastic improvement in performance of around 5% in precision and 10% in recall. We attribute this to the fact that the semantic similarities allow better generalization to previously unknown words in the test set. This leads primarily to a reduction of recall errors,

⁴The deWac corpus supports the training of larger models. However, recall that the runtime is quadratic in the number of clusters, and the optimal number of clusters grows with the corpus size. This leads to long clustering times.

Tokens	Clusters	Precision	Recall	F ₁
Baseline (0/0)		80.9	58.8	68.1
10M	100	85.2	68.1	75.7
10M	200	85.2	66.8	74.9
20M	100	83.0	64.9	72.9
20M	200	86.4	70.1	77.4
50M	200	86.7	69.3	77.0
50M	400	87.3	71.5	78.6
100M	200	85.4	69.4	76.6
100M	400	86.7	76.0	77.8
175M	200	86.2	71.3	78.0
175M	400	87.2	71.0	78.3
175M	600	88.0	72.9	79.8

Table 1: Performance on CoNLL German TestA development set, using HGC as generalization corpus

Tokens	Clusters	Precision	Recall	F ₁
Baseline (0/0)		80.9	58.8	68.1
10M	100	83.5	65.5	73.4
10M	200	84.1	66.0	73.9
20M	100	84.2	66.2	74.1
20M	200	84.1	66.8	74.5
50M	200	85.4	68.9	76.3
50M	400	85.1	68.9	76.1
100M	200	84.9	68.6	75.9
100M	400	84.8	69.1	76.1
175M	200	85.0	69.4	76.4
175M	400	86.0	70.0	77.2
175M	600	85.4	69.3	76.5

Table 2: Performance on CoNLL German TestA development set, using deWac as generalization corpus

but to more robust regularities in the model, which improves precision. The beneficial effect of the generalization corpus increases from 10M tokens to 50M tokens, leading to a total improvement of 6-7% in precision and 12-13% in recall, but levels off afterwards, indicating that no more information can be drawn from the HGC. For all but the smallest generalization corpus size, more clusters improve performance.

The situation is similar, but somewhat different, when we use the deWac corpus (Table 2). For 10M tokens, the improvement is considerably smaller, only 2.5% in precision and 6.5% in recall. However, the performance keeps improving when more data is added. At the size of the HGC (175M tokens), the performance is only about 1% worse in all statistics than for the HGC. As can be seen in Figure 1, the performances for HGC and deWac seem largely to converge. This is a promising result, given that we did not do any cleaning of deWac, since web corpora are cheaper than newswire corpora and can be obtained for a larger range of languages.

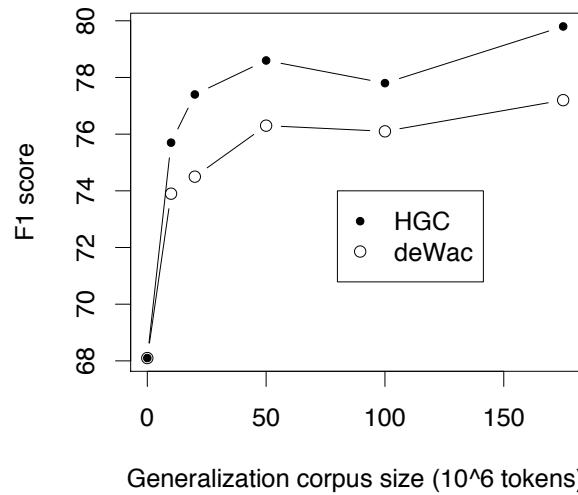


Figure 1: F₁ as function of generalization corpus

Model	Precision	Recall	F ₁
Florian et al. (2003)	83.9	63.7	72.4
Baseline (0/0)	84.5	63.1	72.3
HGC (175M/600)	86.6	71.2	78.2
deWac (175M/400)	86.4	68.5	76.4

Table 3: Comparison to best CoNLL 2003 results for German on the CoNLL TestB test dataset

Finally, Table 3 validates the results for the best HGC and deWac models on the test set (TestB) and compares them to the best CoNLL 2003 shared task system for German (Florian et al., 2003). We see a small decrease of the performance of both systems by about 1% F-Score. Both models substantially outperform the baseline without generalization and Florian et al., a classifier combination system, by 4% and 5% F-Score, respectively. The improvement is mainly due to an 8% increase in recall.

5 Exp. 2: Testing on Out-of-Domain Data

This experiment assesses the performance of our CoNLL-trained German NER system on a different domain, namely the German part of the EUROPARL corpus (Koehn, 2005). EUROPARL consists of the Proceedings of the European Parliament, i.e., corrected transcriptions of spoken language, with frequent references to EU-related NEs. It thus differs from CoNLL both in genre and in domain. We annotated the first two EUROPARL files⁵ with NEs according to the CoNLL guidelines, resulting in an out-of-domain test set of roughly 110,000 tokens.

⁵ep-96-04-{15, 16}; tagging speed ≈ 2000 tokens/h.

Model	Precision	Recall	F ₁
Baseline (0/0)	67.8	47.4	56.0
HGC (175M/600)	78.0	56.7	65.6
deWac (175M/400)	77.0	56.7	65.3

Table 4: Performance on EUROPARL

Results. We tagged the test set with the baseline model and the best HGC and deWac models. The results are shown in Table 4. The performance of the baseline model without generalization is considerably worse than on the in-domain test set, with a loss of about 10% in both precision and recall. We see particularly bad recall for the MISC and ORG classes (34.4% and 46.0%, respectively), which are dominated by terms infrequent in newswire (nationalities and EU organizations and programs).

With semantic generalization, both recall and precision increase by roughly 10% for both HGC and deWac, indicating that corpus quality matters less in out-of-domain settings. We find a particularly marked improvement for the LOC category (deWac: P: 85.5% → 93.5%; R: 53.4% → 71.7%). We attribute this to the fact that location names are relatively easy to cluster distributionally and thus profit most from the semantic generalization step. Unfortunately, the same is not true for the names of EU organizations and programs. Even though the final performance of the models on EUROPARL is still around 10% worse than on the in-domain test data, the comparatively high precision suggests that the systems may already be usable for term extraction or in some semi-automatic setup.

6 Related Work

Rössler (2004) follows a similar motivation to ours by compiling resources with lexical knowledge from large unlabelled corpora. The approach is implemented and evaluated only for the PER(son) category. Volk and Clematide (2001) present a set of category-specific strategies for German NER that combine precompiled lists with corpus evidence. In contrast, Neumann and Piskorski (2002) describe a finite-state based approach to NER based on contextual cues and that forms a component in the robust SMES-SPPC German text processing system. Didakowski et al. (2007) present a weighted transducer-based approach which integrates LexikoNet, a German semantic noun classification with 60,000 entries.

Table 5 compares the performance of these systems on the only category that is available in all systems, namely PER(son). System performance

System	Data	Prec	Rec	F ₁
HGC 175M/600 Rössler (2004)	C	96.2	88.0	92.0
	C	89.4	88.4	88.9
Didakowski et al. (2007)	O	93.5	92.8	93.1
Volk and Clematide (2001)	O	92	86	88.8
Neumann and Piskorski (2002)	O	95.9	81.3	88.0

Table 5: Different German NER systems on category PER (C: CoNLL 2003 test set, O: own test set)

is between 88% and 93% F-Score, with the best results for Didakowski et al. and our system. This comparison must however be taken with a grain of salt. Only our system and Rössler's are evaluated on the same data (CoNLL 2003), while the three other systems use their own gold standards. Still, our HGC model performs competitively with the best systems for German, in particular with respect to precision.

7 Conclusions

We have presented a study on training and evaluating a Named Entity Recognizer for German. Our NER system alleviates the absence of large training corpora for German by applying semantic generalizations learned from a large, unlabelled German corpus. Corpora from the same genre yield a significant improvement already when relatively small. We obtain the same effect with larger web-crawled corpora, despite the higher potential noise. Applied across domains, there is no practical difference between the two corpus types.

The semantic generalization approach we use is not limited to the four-class CoNLL setup. Even though its benefit is to decrease the entropy of the NE classes distribution by conditioning on clusters, and a higher number of NE classes could reduce the size of this effect, in practice the number of clusters is much higher than the number of NER classes. Therefore, this should not be an issue. Generalization can also be combined with any other models of NER that can integrate the class features. The extent to which other systems (like Florian et al., 2003) will improve from the features depends on the extent to which such information was previously absent from the model.

We hope that our results can be helpful to the German NLP community. Our two best classifiers (HGC 175M/600 and deWac 175M/400) as well as the EUROPARL test set are freely available for research at http://www.nlpado.de/~sebastian/ner_german.html.

Acknowledgements Many thanks to Jenny Rose Finkel and Alexander Clark for their support.

References

- Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Journal of Language Resources and Evaluation*, 43(3):209–226.
- Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 59–66, Budapest, Hungary.
- Jörg Didakowski, Alexander Geyken, and Thomas Hanneforth. 2007. Eigennamenerkennung zwischen morphologischer Analyse und Part-of-Speech Tagging: ein automatentheoriebasierter Ansatz. *Zeitschrift für Sprachwissenschaft*, 26(2):157–186.
- Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore.
- Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of the Conference on Natural Language Learning*, pages 168–171, Edmonton, AL.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the MT Summit X*, Phuket, Thailand.
- Günter Neumann and Jakub Piskorski. 2002. A shallow text processing core engine. *Journal of Computational Intelligence*, 18(3):451–476.
- Marc Rössler. 2004. Corpus-based learning of lexical resources for German named entity recognition. In *Proceedings of the Language Resources and Evaluation Conference*, Lisbon, Portugal.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Natural Language Proceedings*, pages 44–49, Manchester, UK.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Conference on Natural Language Learning*, pages 142–147, Edmonton, AL.
- Martin Volk and Simon Clematide. 2001. Learn-filter-apply-forget. Mixed approaches to named entity recognition. In *Proceedings of the 6th International Workshop on Applications of Natural Language for Information Systems*, Madrid, Spain.

Aufbau eines linguistischen Korpus aus den Daten der englischen Wikipedia

Markus Fuchs

Informationswissenschaft

Universität Regensburg

Regensburg, Deutschland

fuchs.markus@gmail.com

Abstract

Die Online-Enzyklopädie Wikipedia bietet eine Fülle an (semantischer) Information, die zur Lösung von Problemstellungen aus verschiedenen Bereichen der Sprach- und Texttechnologie (*Information Retrieval*, Verarbeitung natürlicher Sprache, *Information Extraction*) eingesetzt werden kann. Die Informationen liegen jedoch nur in semi-strukturierter Form in der Wikipedia-eigenen Markup-Sprache vor, dem sogenannten ‐Wikitext‐.

Dieser Artikel stellt ein System vor, das auf Basis der Daten der (englischen) Wikipedia ein linguistisches Korpus erstellt und darin auch Wikipedia-spezifische strukturelle Daten und Metadaten wie Artikelverweise, Kategorienzugehörigkeit oder zwischensprachliche Verweise speichert. Da alle Daten in einer relationalen Datenbank abgelegt werden, ist ein einfacher und sehr effizienter Zugriff mit umfassender Suchfunktionalität möglich. Es wird unter anderem gezeigt, wie sich die drei häufigsten Untersuchungsmethoden von Korpora (Konkordanz, Frequenzlisten und Kollokationen) mit dem hier beschriebenen System realisieren lassen.

1 Einleitung

Eine neue Art der semantischen Ressource findet in den letzten Jahren verstärkt Anwendung in der Verarbeitung natürlicher Sprache: Sogenannte kolaborativ erstellte Wissensressourcen (KWR) bieten gegenüber traditionellen Ressourcen wie WordNet oder GermaNet einige Vorteile, die sie für den Einsatz bei vielen Problemstellungen interessant machen (Zesch et al., 2008):

Zum einen sind sie (zumeist) frei verfügbar und zum anderen kann durch die große Zahl an freiwilligen Mitarbeitern ein hoher Grad an Abdeckung und Aktualität erreicht werden.

Die Wikipedia als der wohl bedeutendste Vertreter der KWRs bietet durch ihren enzyklopädischen

Aufbau, ihre dichte Verweisstruktur und ihre Multilingualität eine Fülle an (semantischer) Information und wurde bereits für eine Vielzahl verschiedener Studien eingesetzt:

Gabrilovich und Markovitch (2009) machen sich die Tatsache zu Nutze, dass jeder Wikipedia-Artikel einen Begriff beschreibt, und gründen darauf ein Verfahren zur Berechnung der semantischen Verwandtschaft. Ruiz-Casado u.a. (2005) vergleichen die Artikeltexte mit den Glossierungen potenziell bedeutungsgleicher WordNet-Synsets, um Wikipedia-Einträge automatisiert auf die entsprechenden Konzepte von WordNet abzubilden.

In der Wikipedia kommen drei verschiedene Arten von Verweisen vor: Links auf andere Artikel werden z. B. von Ito u.a. (2008) eingesetzt. Sie nutzen deren Kookkurrenzen, um ein System zur Bestimmung der semantischen Verwandtschaft zu erstellen.

Kategorien-Links werden in der Wikipedia verwendet, um mehrere Artikel bzw. mehrere Kategorien in einer Kategorie zusammenzufassen. Ponzetto und Strube (2007) analysieren beispielsweise die Syntax der Titel verknüpfter Kategorien, um deren semantische Beziehung zu bestimmen.

Durch zwischensprachliche Verweise können Artikel mit gleichem Inhalt unterschiedlicher Sprachversionen der Online-Enzyklopädie miteinander verknüpft werden. De Smet und Moens (2009) verwenden dies in einem Verfahren, mit dem sich bestimmen lässt, ob zwei Nachrichtenmeldungen in verschiedenen Sprachen über das gleiche Ereignis berichten.

2 Vergleichbare Arbeiten

Für den Zugriff auf die Wikipedia(WP)-Daten werden sogenannte ‐Datenbank-Backup-Dumps‐ angeboten.¹ Dabei handelt es sich um eine XML-Datei, die in mehr oder weniger regelmäßigen Abständen

¹<http://dumps.wikimedia.org/enwiki/>

(alle 1-3 Wochen) für alle Sprachversionen erstellt wird und alle zu diesem Zeitpunkt bestehenden Artikel enthält. Die Texte der einzelnen Artikel selbst sind im Wikipedia-eigenen Wikitext-Format gespeichert.

Um strukturiert auf die Daten zugreifen zu können, muss der Wikitext erst geparsst werden. Deshalb gibt es mittlerweile einige Projekte, bei denen dieser Schritt bereits erledigt ist, wie z. B. die *Java Wikipedia Library*(JWPL) (Zesch et al., 2008), den *Semantically Annotated Snapshot of Wikipedia*(SW1) (Atserias et al., 2008) oder das *WaCkypedia_EN*-Korpus (Baroni et al., 2008). Keines dieser Systeme bzw. Ressourcen bietet allerdings einen effizienten und umfassenden Zugriff auf die in der Wikipedia enthaltene Informationsmenge dergestalt, dass komplexe Suchanfragen formuliert werden können.

So wurden die Wikipedia-Artikeltexte bei SW1 zwar um eine Vielzahl an linguistischen Annotationsdaten angereichert, jedoch fehlen viele wichtige Wikipedia-spezifische Daten wie zwischen-sprachliche Verweise oder die Kategorienzugehörigkeiten. Zudem sind die Daten in mehreren Textdateien im proprietären “Multitag”-Format gespeichert, sodass eine Verwendung nicht unmittelbar möglich ist.

Bei der JWPL hingegen lassen sich die Daten über eine umfangreiche Programmierschnittstelle abfragen. Hier sind aber wiederum keine zusätzlichen Annotationsdaten enthalten. Darüber hinaus sind Suchmöglichkeiten auf eher strukturelle Daten beschränkt (z. B. “Gib alle Artikel aus, die eine bestimmte Anzahl an einkommenden Verweisen haben.”).

Das *WaCkypedia_EN*-Korpus ist eines der Korpora aus dem *WaCky*-Projekt, bestehend aus den Artikeltexten der englischen Wikipedia. Die Daten sind in einem XML-Format gespeichert, das von der *IMS Open Corpus WorkBench* gelesen und indiziert werden kann. Die Artikeltexte wurden um linguistische Annotationsdaten erweitert. Jedoch sind keine Wikipedia-spezifischen Daten wie etwa die Artikelverweise enthalten.²

3 System zur Erstellung eines Wikipedia-Korpus

Um die Nachteile der oben beschriebenen Projekte zu umgehen, muss das hier vorgestellte System die

folgenden Anforderungen erfüllen:

Da der Datenbestand der Wikipedia ständig anwächst, muss das System gut skalieren können. Des Weiteren sollen darin sowohl Wikipedia-spezifische als auch linguistische Daten enthalten sein, damit das Korpus für möglichst viele wissenschaftliche Fragestellungen verwendet werden kann. Und schließlich soll das Speicherformat zum einen so gestaltet sein, dass sich zusätzliche Daten (z. B. weitere linguistische Annotationen) auch nachträglich leicht hinzufügen lassen. Zum anderen sollen die Daten leicht in ein Standardformat exportiert werden können.

3.1 Plattform und Architektur

Das System wurde in C++ mit plattformunabhängigen Standardbibliotheken programmiert. Als Datenspeicher wird eine PostgreSQL-Datenbank verwendet.

Dadurch kann die in dem relationalen Datenbank-Management-System (RDBMS) implementierte Indizierungs- und Suchfunktionalität auch bei der Korpus-Abfrage verwendet werden. Und über die Abfragesprache SQL sind auch komplexere Abfragen und Mustersuchen möglich. Weiterhin lassen sich auch nachträglich weitere Daten hinzufügen, indem die neuen Datentabellen über Fremdschlüssel mit den bestehenden verknüpft werden. Außerdem erleichtert das stark strukturierte Speicherformat einer relationalen Datenbank den Export der Daten in ein anderes Format (z. B. XCES).

Die einzelnen Artikel-Datensätze des Wikipedia-Daten-Dumps lassen sich unabhängig voneinander verarbeiten und speichern. Deshalb wurde die Verarbeitung parallelisiert und auf drei Programme (*WikiServer*, *WikiCorpusClient* und *CorpusServer*) verteilt, die auf unterschiedlichen Rechnern laufen können. Dadurch lässt sich die Erstellung des Korpus stark beschleunigen.

3.2 Wikitext-Parser

Der erste Verarbeitungsschritt ist das Parsen des Wikitext-Markups. Um größtmögliche Kompatibilität mit dem Original-Parser zu erreichen, wurden die dort verwendeten Ersetzungs-Algorithmen übernommen und in einer modular aufgebauten, objektorientierten und dadurch leichter zu wartenden C++-Bibliothek implementiert.³

²Eine ausführlichere Darstellung vergleichbarer Arbeiten findet sich in Fuchs (2009).

³Für eine nähere Beschreibung des Parsers sei auf Fuchs (2009) verwiesen.

Aus dem vom Parser erstellten Parsebaum wird zuerst der reine Artikeltext (ohne Formatierungen) extrahiert. Zusätzlich werden aber auch alle Verweistypen und Hervorhebungen (fett, kursiv) gespeichert.

3.3 Lexikalische Verarbeitung

Zur lexikalischen Verarbeitung des reinen Artikeltextes wurde der FoxTagger (Fuchs, 2007) verwendet. Da der Part-of-Speech-Tagger bereits sowohl eine Satzgrenzen-Erkennung als auch die Tokenisierung integriert hat, sind für diese notwendigen Verarbeitungsschritte keine weiteren Programme nötig.

Als Lexikon verwendet der Tagger nicht wie sonst üblich ein aus dem Trainingskorpus erstelltes Lexikon, sondern die Index-Dateien von WordNet für die offenen Wortklassen und eine von Hand erstellte Liste für die geschlossenen Wortklassen.

Fuchs (2007) hat in Untersuchungen festgestellt, dass diese Konfiguration einen positiven Effekt auf die Genauigkeit und Robustheit des Taggers hat. Deshalb ist davon auszugehen, dass FoxTagger auch auf dem Datenbestand der Wikipedia sehr gute Ergebnisse liefert, obwohl sich die darin enthaltenen Texte vermutlich stark vom Trainingskorpus unterscheiden. Da es leider keinen Gold-Standard für die Part-of-Speech(PoS)-Annotation der Wikipedia-Texte gibt, konnte diese Vermutung nicht verifiziert werden.

In dem PoS-Tagger ist außerdem eine morphologische Analyse implementiert, die für jedes Token alle für die zugewiesene Wortart möglichen Lemmata ausgibt.

3.4 (Ko-)Okkurrenz-Analyse

Die Untersuchung von Frequenzlisten und Kollokationen sind typische Verwendungen von Korpora. Um beide leicht aus dem Wikipedia-Korpus abfragen zu können, werden die Vorkommenshäufigkeiten der Terme in jedem Artikel gezählt und in der Datenbank abgespeichert. Über Aggregatfunktionen des RDBMS können die Häufigkeitswerte der einzelnen Artikel dann summiert werden.

Ebenso werden bei der Erstellung des Korpus auch die direkten Kookkurrenzen (*surface cooccurrences*) über ein asymmetrisches Kontextfenster mit einer Größe von vier Tokens nach rechts gezählt (Evert, 2008).

Bei der Zählung von Kookkurrenzen und der Berechnung der Assoziationsmaße überlagern sich die

verschiedenen Bedeutungen der Terme. Die Filterung nach Part-of-Speech erlaubt die Auftrennung in begrenztem Umfang. Eine klare Trennung wäre nur bei einer Bedeutungsannotation möglich.

Mihalcea (2007) hat gezeigt, dass sich die Artikelverweise als Bedeutungsannotation von deren Ankertexten interpretieren lassen. Aus diesem Grund erfolgt die Zählung der Kookkurrenzen auf zwei Artikelversionen: Bei der einen werden die Tokens des Ankertextes gezählt und bei der anderen die Artikelverweise selbst. Damit lassen sich auch die Kollokationen von “Wikipedia-Begriffen” ermitteln, wenn auch auf einer kleineren Datenbasis.

3.5 Speicherformat

Die Daten des Korpus sind auf mehrere Datenbank-Tabellen verteilt. Kernstück ist die Tabelle *corpus.tokens*. Sie enthält für jedes Token des Korpus eine Tabellenzeile mit den folgenden Daten: die Wortform des Tokens (*surface*), seine Grundform (*lemmata*), Wortart (*part_of_speech*) und Position im Korpus (*token_pos*); die ID des Artikels, in dem es vorkommt (*rev_id*); die Nummer des Satzes, in dem es vorkommt (*sentence_id*), sowie die Position in diesem Satz (*pos_in_sentence*) und schließlich, ob das Token fett oder kursiv formatiert ist (*is_emphasized*). Zusätzlich ist für jedes Token, das im Ankertext eines Links vorkommt, der Name des Artikels gespeichert, auf den verwiesen wird (*links_to_article*).

In weiteren Tabellen sind alle Artikel-, Kategorien- und zwischensprachlichen Verweise gespeichert (*article_links*, *article_categories* und *article_iw_links*). Die *corpus_articles*-Tabelle enthält für jeden Artikel einen eigenen Datensatz, in dem dessen Titel, ID (*rev_id*) und Informationen über die Anzahl der darin enthaltenen Tokens gespeichert sind. Die im vorherigen Abschnitt erwähnten (Ko)Okkurrenz-Häufigkeiten für jeden Artikel sind in den beiden Tabellen *article_term_frequencies* und *article_cooccurrence_frequencies* gespeichert. Zusätzlich enthält die Datenbank die beiden Tabellen *corpus_term_frequencies* und *corpus_cooccurrence_frequencies*, in denen die Häufigkeitswerte für das Gesamtkorpus bereits summiert wurden.

Wie die im Korpus enthaltenen Daten für typische korpusbasierte Studien (Frequenzlisten, Kollokationen, Konkordanz) verwendet werden können, zeigt der nächste Abschnitt.

4 Anwendungsmöglichkeiten

Da die Termhäufigkeiten des Gesamtkorpus bereits in einer eigenen Tabelle gespeichert sind, ist die Abfrage von Frequenzlisten sehr einfach. Dabei können die Algorithmen der Datenbank zur Sortierung und Filterung verwendet werden. Da die Vorkommenshäufigkeiten der Terme auch für jeden Artikel in der Datenbank abgelegt sind, lassen sich auch Frequenzlisten für Subkorpora (z. B. für alle Artikel einer Kategorie) sehr leicht erstellen.

Interessieren nur die Frequenzwerte für eine begrenzte Anzahl Terme, können diese in einer eigenen Tabelle abgelegt und zur Filterung verwendet werden.

Um Kollokationen im Sinne von wiederkehrenden und vorhersagbaren Wortkombinationen (Evert, 2008) aus dem Korpus zu extrahieren, werden verschiedene Assoziationsmaße verwendet. Zur Berechnung werden die Vorkommenshäufigkeiten der beiden Terme und deren Kookkurrenzen sowie die Gesamtzahl aller Tokens des Korpus benötigt. Auch hier lassen sich diese Werte wieder direkt aus den entsprechenden Tabellen (*corpus_term_frequencies* für die Termfrequenzen bzw. *corpus_cooccurrence_frequencies* für die Kookkurrenzhäufigkeiten) abfragen. Wie oben bereits erwähnt, können die Artikelverweise als Bedeutungsannotation interpretiert und auch deren Kollokationen ermittelt werden.

Es ist angedacht, die Assoziationswerte für die gängigsten Assoziationsmaße (Evert, 2008) zu berechnen und in der Datenbank abzuspeichern.

Zur Erstellung einer Konkordanz-Ansicht können alle Positionen, an denen das gesuchte Knotenwort auftritt, und deren Kontext über einen Tabellenindex abgefragt werden.

Auch hier ist es wieder möglich, die Artikelverweise zu nutzen. Somit lässt sich auch die Konkordanz einer bestimmten Wortbedeutung darstellen.

5 Ausblick

Für das Korpus ist ein Webinterface (<http://www.wikicorpus.com>) in Vorbereitung, über das sich die Daten auch ohne SQL abfragen lassen. Dort soll es dem Benutzer dann auch möglich sein, das Korpus um eigene Annotationen zu erweitern oder Listen in der Datenbank anzulegen, auf die dann bei Abfragen referenziert werden kann.

Ebenso ist für die nächste Version des Korpus-Systems geplant, noch mehr Daten aus der Wiki-

pedia zu extrahieren und in der Datenbank abzuspeichern. Dazu zählen z. B. die Informationen aus Infoboxen oder über den Aufbau des Artikels (Sktionen, Paragraphen).

Der Datenbestand wird in Zukunft automatisch über die Wikipedia-Daten-Dumps laufend aktualisiert.

Der Großteil der einzelnen Verarbeitungsschritte zur Erstellung des Korpus ist sprachunabhängig. Deshalb ist eine Verwendung des Korpus-Systems für weitere Sprachen möglich und angedacht.

References

- J. Atserias, H. Zaragoza, M. Ciaramita, and G. Attardi. 2008. Semantically annotated snapshot of the english wikipedia. In *LREC*.
- M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. 2009. A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.
- C. Biemann, S. Bordag, G. Heyer, and U. Quasthoff. 2004. Language-independent methods for compiling monolingual lexical data. In *Proceedings of CICLING 2004*, pages 217–228.
- W. De Smet and M.-F. Moens. 2009. Cross-language linking of news stories on the web using interlingual topic modelling. In *SWSM '09: Proceeding of the 2nd ACM workshop on Social web search and mining*, pages 57–64. ACM.
- L. Denoyer and P. Gallinari. 2006. The wikipedia xml corpus. *SIGIR Forum*, 40(1):64–69.
- S. Evert. 2008. *Corpora and collocations*, chapter 58, pages 1212–1249. Walter de Gruyter.
- W. M. Francis and H. Kucera, 1964. *Brown Corpus. Manual of Information*.
- M. Fuchs. 2007. Automatische extraktion und annotation formaler textmerkmale. Diplomarbeit, Hochschule Regensburg.
- M. Fuchs. 2009. Aufbau eines wissenschaftlichen textcorpus auf der basis der daten der englischsprachigen wikipedia. Masterarbeit, Hochschule Regensburg.
- E. Gabrilovich and S. Markovitch. 2009. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Resources*, 34(1):443–498.
- M. Ito, K. Nakayama, T. Hara, and S. Nishio. 2008. Association thesaurus construction methods based on link co-occurrence analysis for wikipedia. In *CIKM 2008*, pages 817–826.

- A. Mehler, R. Gleim, A. Ernst, and U. Waltinger. 2008. Wikidb: Building interoperable wiki-based knowledge resources for semantic databases. *Sprache und Datenverarbeitung. International Journal for Language Data Processing*, 32(1):47–70.
- R. Mihalcea. 2007. Using wikipedia for automatic word sense disambiguation. In *HLT-NAACL*, pages 196–203. The Association for Computational Linguistics.
- S. P. Ponzetto and M. Strube. 2007. Deriving a large scale taxonomy from wikipedia. In *AAAI*, pages 1440–1445.
- M. Ruiz-Casado, E. Alfonseca, and P. Castells. 2005. Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. In *AWIC*, pages 380–386.
- F. Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.
- T. Zesch, C. Müller, and I. Gurevych. 2008. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of LREC'08*, pages 1646–1652.

Detecting Vagueness in Two Languages and Two Domains

Viola Ganter

HITS gGmbH

Heidelberg

Germany

viola.ganter@h-its.org

Abstract

We describe a system for the automatic detection of weasel worded subclauses in English and French. We extract already annotated sentences from Wikipedia articles and use n-gram- and POS-based features for weasel word detection. We successfully apply the English model to biomedical, hedge-annotated data.

1 Introduction

Recent years have seen a surge in research on the detection of speculative statements (*hedges*). However, little research has been done outside the biomedical domain or outside the genre of scientific papers. Furthermore, to our knowledge, there have been no computational studies in languages other than English. On the other hand, many NLP systems would benefit from the detection of vague statements. Question-answering and ontology-extraction systems need to be able to differentiate facts from speculation. Automatic essay-correction and -writing aids could benefit from the detection of vague references, and research in deception detection needs to identify agent underspecification. To this end, Ganter & Strube (2009) have investigated the detection of weasel words in Wikipedia articles, using a combination of unigrams and heuristic, shallow syntactic patterns.

In this paper, we first propose a definition of weasel words and contrast it to hedges. We then investigate different feature sets and their benefits for weasel word detection. Finally, we show that the task can be transferred to other languages and domains by applying our system to French Wikipedia articles and a biomedical hedge corpus.

2 Related Work

Research on hedge detection in NLP has been focused almost exclusively on the biomedical domain. Light et al. (2004) present a study on annotating hedges in biomedical documents. They show that the phenomenon can be annotated tentatively reliably by non-domain experts when using a two-way distinction. Medlock & Briscoe (2007) develop a weakly supervised, unigram-based system for hedge classification in the biomedical domain. Their best system results in a 0.76 precision/recall break-even-point (BEP). Szarvas (2008) extends their work to n-grams. He also applies his method to (slightly) out of domain data and observes a considerable drop in performance. Kilicoglu & Bergler (2008) are first to consider the effect of syntactic patterns in hedge detection, yielding an F-score of 85% when trained and tested on the corpus created by Medlock & Briscoe (2007). They also participate in the BioNLP Shared task 2009 (Kilicoglu & Bergler, 2009), reaching first place with an F-score of 25.27% for the detection of speculative language. This low F-score partially relates to the setup of the task, which involved event extraction as a basis to hedge detection, but on the other hand the authors also point out “the lack of a standardized notion of speculation among various corpora” (p. 124).

3 Weasels and Hedges

There is little consistency in the usage of the term *hedge*. Hedges were introduced by Lakoff as *words whose meaning implicitly involves fuzziness* (Lakoff, 1973). His examples for hedges include *sort of* and *often*, but also *par excellence* and *really* - the latter of which being particularly atypical for the contemporary use of the term. Crompton (1997) considers different studies of hedges and proposes definition (1) below.

Hedging in scientific context strongly differs from hedging in political speeches. It is a required means to pose uncertain suppositions, whereas Jason (1988) focuses on hedging in political context, where it has an evasive, euphemistic nature.

Weasel words is a colloquial term and, thus, similarly ill-defined. Jason (1988) describes weasel words as a means to political hedging, and defines them as an *expression denoting qualifying phrases and detensifiers (sort of or rather)*. The Wikipedia style guidelines describe weasel words as *... words that, whilst communicating a vague or ambiguous claim, create an impression that something specific or meaningful has been said*.

We propose the following distinction:

1. A **hedge** is an item of language which a speaker uses to explicitly qualify his/her lack of commitment to the truth of a proposition he/she utters. (Crompton, 1997)
2. A **weasel word** is an item of language which a speaker uses to disguise his/her lack of commitment to the truth of a proposition he/she utters.

Both hedges and weasel words are means for *hedging*, which however, can be divided into explicit and evasive hedging. What hedges and weasel words have in common, then, is the speaker's lack of commitment to the truth of the statement they appear in.

4 Data

Contributors to Wikipedia are advised to avoid *weasel words*, as they are not in line with Wikipedia's policy of a neutral point of view¹. To this end, a number of maintenance tags can be placed into articles with weasel words (hereafter referred to as *weasel tag*). We considered all those tags to be weasel tags, that were either listed by Wikipedia as tags to improve weasel worded statements (`{{weasel word}}` , `{{who}}`) or were a redirect of one of these tags (i.e. tags that were directly linked to and used as equivalents of one of the listed tags). However, we excluded the `{{weasel}}` tag, as it is generally used to tag whole paragraphs or even articles as weasel worded, while the other tags are placed directly

¹http://en.wikipedia.org/wiki/Wikipedia:Guide_to_writing_better_articles

into a sentence. To collect training data, we extracted subclauses containing weasel tags from POS tagged Wikipedia articles (using Wikipedia dumps² from the years 2007-2009 and the Tree-Tagger (Schmid, 1997)), subclauses in this case being defined as the string between two punctuation marks. For each tagged subclause we found, we chose one random subclause from an article that did not contain any weasel tag in order to collect negative examples. The reason for this choice of negative examples related to the fact that weasel worded sentence were often surrounded by other, untagged weasel worded sentences. In some cases articles were marked as generally weasel worded and the weasel-word tag was just used to emphasize some example sentences. In other cases, the occurrence of a weasel-worded expression was tagged once, but left untagged in succeeding sentences. By randomly choosing negative instances from completely untagged pages, we intended to minimize the number of false negatives in the training set, although it should be noted that this still did not completely exclude the possibility of false negatives. We extracted a total of 1008 subclauses for training and 110 each as development and test data.

5 Experiments

5.1 Method

We created eight different feature sets: Following Medlock & Briscoe (2007) and Szarvas (2008), we extracted *unigrams* and *bigrams* from the training data to create feature sets. However, we expected these features not to be informative enough to characterize weasel words. Consider the following examples:

- 1 a) *It is said* (weasel)
 b) *They were considered* (weasel)
 c) *They were generally considered* (weasel)
- 2 a) *Some linguists* (weasel)
 b) *Some church representatives* (weasel)
 c) *The church representatives* (non-weasel)

Example 1a) is a weasel worded phrase, although none of its containing unigrams (*it*, *is*, or *said*) is a weasel word. While bigrams might capture this, we did not expect them to capture the similarity between 1b) and 1c). To meet this problem, we also extracted k-skip-bigrams, short *skipgrams*. Given a sequence of words $w_1 \dots w_n$ we

²<http://download.wikipedia.org/>

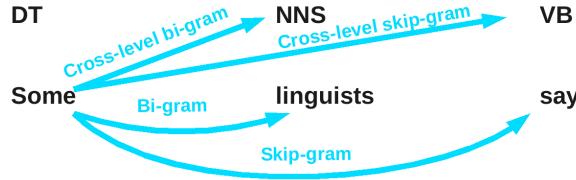


Figure 1: Feature extraction: bigrams, skip-grams, cross-level bigrams, and cross-level skip-grams

define skip-grams as the set of word pairs $\{(w_i, w_j) \mid i \in \{1 \dots n-1\} \wedge j \in \{i+1 \dots n\}\}$.

However, we expected none of the above-mentioned features to capture the similarity between 1a) and 1b) or c). The weasel-wordedness of these examples seems to be beyond the word level. To capture this, we extracted *POS tags* as one feature set and, analogously to the string-based features, also extracted *POS bigrams* and *POS skip-grams*.

In many cases, the characteristic of weasel words can be summarized in a combination of words and POS tags. For instance, 2a) and b) can be characterized by the pattern <Some . * NNS>. In contrast, <DT . * NNS> is not specific enough, as it would also capture example 2c). In order to capture these relations, we extracted what we will call *cross-level bigrams* and *cross-level skip-grams* (see Figure 1 for illustration). Given a sequence of pairs of words and POS tags $(w_1, p_1) \dots (w_n, p_n)$, we define a cross-level bigram as any element of the set $\{(w_i, p_{i+1}), (p_i, w_{i+1}) \mid i \in \{1 \dots n-1\}\}$ and a cross-level skip-gram as any element of the set $\{(w_i, p_j), (p_i, w_j) \mid i \in \{1 \dots n-1\} \wedge j \in \{i+1 \dots n\}\}$. Similarly to Ganter & Strube (2009), we extracted only those features that occurred before a weasel tag. To further eliminate weak features, we performed feature selection using the attribute subset evaluator and best-first search algorithm (performing greedy hill-climbing with backtracking), both implemented in the *Weka* machine learning toolkit (Witten & Frank, 2005). Finally, we combined all (selected) features to a ninth feature set.

5.2 Results

For the task of automatically identifying weasel worded subclauses we applied a support vector machine (SVM), Naïve Bayes (NB), and a decision tree (J48) classifier, all implemented in *Weka*. Table 1 shows the performance of the three classifiers using the different feature sets. As the

precision-recall ratio was similar over all three classifiers, we only report the support vector machine's detailed results. All classifiers performed significantly better than chance (with $p=0.1$ for the bigram-based model, $p=0.01$ for all others using a paired t-test). Surprisingly, POS-based and POS bigrams-based classification performed best with a reasonably high precision considering the coarseness of POS tags. This indicates that weasel words have common characteristics at the POS level that distinctly differentiate them from non-weasel worded language.

Purely word level based features consistently yielded high precision but low recall over all classifiers. This indicates a variety of weasel words on the word level that could not be captured by the training data.

	P_{SVM}	R_{SVM}	F_{SVM}	F_{NB}	F_{J48}
Unigr.	0.90	0.59	0.71	0.76	0.73
POS tags	0.71	0.69	0.70	0.77	0.77
Bigr.	0.83	0.57	0.67	0.53	0.57
POS bigr.	0.75	0.74	0.75	0.77	0.73
C.-l. bigr.	0.76	0.60	0.67	0.69	0.67
Skip-gr.	0.94	0.57	0.71	0.76	0.72
POS skip-gr.	0.78	0.72	0.75	0.74	0.72
C.-l. skip-gr.	0.83	0.66	0.73	0.67	0.67
Combined	0.91	0.66	0.76	0.74	0.74

Table 1: Performance of SVM, NB, and J48 using different features sets (unigrams, part-of-speech tags, bigrams, part-of-speech bigrams, cross-level bigrams, skip-grams, part-of-speech skip-grams, cross-level skip-grams, and the combination of all)

Finally, combining all feature sets yielded the best performance for support vector machine classification, yet it did not differ significantly from other models.

5.3 Experiments on French Data

The French Wikipedia does not contain direct equivalents to each of the English weasel tags. In fact, our French collection of weasel tags consisted only of {{qui}}, {{combien}}, and their redirects (see section 4). Both of these tags are used in the same manner as the English {{who}} tag (which itself is a weasel tag). It should be pointed out that this tag is commonly used to mark underspecified subjects and does not capture occurrences of *perhaps* or *probably*. Common contexts for the French weasel tag are *On raconte* (you/some tell), *certains conservateurs* (certain conservatives), or *est considérée* (is considered). We extracted subclauses in the same way

as for the English corpus, using Wikipedia dumps from the year 2009. The corpus comprised of 462 subclauses for training and each 52 subclauses for testing and development data. Again, we used the TreeTagger for POS tagging. Table 2 shows the results for the different methods when applied to French. All classifiers performed significantly better than chance ($p=0.01$).

	P_{SVM}	R_{SVM}	F_{SVM}	F_{NB}	F_{J48}
Unigr.	0.91	0.73	0.81	0.78	0.79
POS tags	0.71	0.85	0.77	0.70	0.72
Bigr.	0.82	0.35	0.49	0.49	0.44
POS bigr.	0.82	0.69	0.75	0.72	0.71
C.-l. bigr.	0.77	0.65	0.71	0.69	0.68
Skip-gr.	0.72	0.50	0.59	0.56	0.54
POS skip-gr.	0.75	0.81	0.78	0.69	0.75
C.-l. skip-gr.	0.81	0.65	0.72	0.71	0.72
Combined	0.79	0.73	0.76	0.69	0.81

Table 2: Performance on French data

Unlike on the English data, the bigram-based method yields a higher precision when trained on French data than the skip-grams, which might relate to the fact that qualifiers are often placed behind the noun in French, as in *plusieurs observateurs politiques*. Thus, the classifier learns the same bigram (plusieur observateurs) from both, a noun phrase with and without qualifier. In contrast, English qualified noun phrases do not have common bigrams with their qualifierless version (*Some linguists* vs. *Some well-known linguists*).

5.4 Hedge Detection on Biomedical Data

To investigate how similar weasel words are to hedges, we applied the English models on a biomedical hedge corpus (Medlock & Briscoe, 2007). Apart from the biomedical hedge corpus created with their semi-supervised model, they also annotated a small set of sentences manually. The manually annotated data set contains more fine-grained annotation, including the hedge trigger, the scope of the hedge and the topic of speculation. As we classify subclauses, we chose to test our system on this data set. We divided the data into subclauses, considering only those subclauses to contain hedges that contained the hedge trigger, ignoring hedge scope. As we did for the Wikipedia data, for each tagged subclause we extracted one untagged subclause, resulting in a balanced data set of 612 subclauses. Table 3 shows the results.

	P_{SVM}	R_{SVM}	F_{SVM}	F_{NB}	F_{J48}
Unigr.	0.73	0.37	0.49	0.67	0.47
POS tags	0.71	0.73	0.72	0.76	0.76
Bigr.	0.77	0.40	0.53	0.42	0.50
POS bigr.	0.73	0.68	0.71	0.71	0.70
C.-l. bigr.	0.78	0.56	0.65	0.61	0.65
Skip-gr.	0.83	0.32	0.46	0.53	0.52
POS skip-gr.	0.74	0.71	0.73	0.71	0.71
C.-l. skip-gr.	0.77	0.58	0.66	0.65	0.63
Combined	0.73	0.51	0.60	0.72	0.57

Table 3: Performance on biomedical data

The performance of all word-based related methods dropped by an average of 13%. Considering that the biomedical data differs from Wikipedia articles not only in the domain, but also in genre, this was not surprising. As precision was still above 70% for all classifiers, this indicates that there is a considerable lexical overlap between weasel words in Wikipedia articles and hedges in biomedical data. The performance of POS-related methods showed no decrease in performance.

6 Conclusions

We have built a system to detect weasel worded subclauses in Wikipedia articles and transferred this system to French. We have applied the English model to a hedge-annotated corpus from the biomedical domain, showing how similar weasel words and hedges are, even when domain *and* genre differ. For both English and the biomedical hedge corpus, POS-based features performed best, yielding not only a high recall but considerably high precision considering the coarseness of POS tags. This shows that hedges are not purely lexically defined.

Acknowledgments. This work has been partially funded by the European Union under the project Judicial Management by Digital Libraries Semantics (JUMAS FP7-214306) and by the Klaus Tschira Foundation, Heidelberg, Germany.

References

- Crompton, Peter (1997). Hedging in academic writing: Some theoretical problems. *English for Specific Purposes*, 16:271–287.
- Ganter, Viola & Michael Strube (2009). Finding hedges by chasing weasels: Hedge detection using Wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Singapore, 2–7 August 2009, pp. 173–176.
- Jason, Gary (1988). Hedging as a fallacy of language. *Informal Logic*, X3.
- Kilicoglu, Halil & Sabine Bergler (2008). Recognizing speculative language in biomedical research articles: A linguistically motivated perspective. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, Columbus, Ohio, 19 June 2008, pp. 46–53.
- Kilicoglu, Halil & Sabine Bergler (2009). Syntactic dependency based heuristics for biological event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, Boulder, Colorado, 5 June 2009, pp. 119–127.
- Lakoff, George (1973). Hedges: A study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic*, 2:458–508.
- Light, Marc, Xin Ying Qiu & Padmini Srinivasan (2004). The language of Bioscience: Facts, speculations, and statements in between. In *Proceedings of the HLT-NAACL 2004 Workshop: Biolink 2004, Linking Biological Literature, Ontologies and Databases*, Boston, Mass., 6 May 2004, pp. 17–24.
- Medlock, Ben & Ted Briscoe (2007). Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 23–30 June 2007, pp. 992–999.
- Schmid, Helmut (1997). Probabilistic Part-of-Speech tagging using decision trees. In Daniel Jones & Harold Somers (Eds.), *New Methods in Language Processing*, pp. 154–164. London, U.K.: UCL Press.
- Szarvas, György (2008). Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, Ohio, 15–20 June 2008, pp. 281–289.
- Witten, Ian H. & Eibe Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco, Cal.: Morgan Kaufmann.

Transliteration among Indian Languages using WX Notation

Rohit Gupta

LTRC

IIT, Hyderabad

India

rohit@research.iit.ac.in

Pulkit Goyal

Information Technology

IIIT, Allahabad

India

Sapan Diwakar

Information Technology

IIIT, Allahabad

India

{pulkit, sapan}@daad-alumni.de

Abstract

In this paper, we propose an algorithm to transliterate between several Indian languages. The main aim of the algorithm is to assist in the translation process by providing efficient transliteration. This algorithm works on Unicode transformation format of an Indian language. It then transliterates it into the Unicode transformation format of the target language. It does no sort of bilingual dictionary lookup of the word. It can be used to transliterate nouns (e.g. named entities) in the translation process as well as for transliterating some text into other language which is more suitable for the reader.

1 Introduction

With the advances in technology and availability of information in electronic format everywhere, it becomes important to provide this information to people as and when needed as well as in their native language. This calls for the development of a tool that can translate this information efficiently.

The translation process comprises of several steps, one of which is transliteration. By transliteration, we refer to the process of transcribing letters or words from one script to another script. In transliteration, word pronunciation is usually preserved. In some cases, it can also be modified according to its pronunciation in target language. Its main aim is to present the word in the destination language's script such that it is readable by the readers of the destination language (Surana and Singh, 2008).

The use of translation is even more necessary in a country like India that has immense diversity. There are different people who speak different languages in different regions of the country. Moreover, most of these languages have different scripts. Thus the application of translation is huge in India. Also, Indian languages are used by many people

across the globe. Hindi, the most common of all the Indian languages is used by more than four hundred million people followed by Bengali (83m), Telugu (74m), Marathi (72m), Tamil (61m), Urdu (52m), Gujarati (46m), Kannada (38m) and Malayalam (33m) (Wikipedia, 2010).

We aim at providing an efficient algorithm for the transliteration process that is used to convert nouns (or other words) that are not present in the bilingual dictionary of the source language to the target language. Such software has other utilities as well when used as a standalone tool. One such utility of such software is in building an interface for users wherein they can type in an Indian Language using the more familiar QWERTY keyboard.

The idea is to allow users to type Roman letters and have them automatically transliterated into Indian Language. This is not as simple as it occurs because there is no direct mapping between Roman letters and Indian Language letters. There may be several combinations of characters which produce a single character in Indian Language or may produce vowel. The mapping that we have used in our work, is a more constrained and provides a rule set for writing a particular Indian Script in Roman letters which can then be converted into the Indian Script. This intermediate representation (known as WX notation explained in a greater detail later in the paper) also provides a way to convert the Indian Languages into one another considering that the phonetic pronunciation of the words in WX notation does not change with different scripts. This assumption is simplifying as well as holds true in most of the cases for Indian Languages. Our approach revolves around this concept of WX notation and inter-conversions between UTF notation of language to its WX notation and then from WX to UTF of the target language.

The paper is structured as follows. In Section 2, we briefly discuss the previous work carried out in this field. In Section 3, we describe our

methodology which is subdivided into three main modules as described in Sections 3.1, 3.2 and 3.3.

2 Previous Research

There have been several researches carried out in this area. Janarthanam, Sethuramalingam and Nal-lasamy (2008) proposed an algorithm that employs grapheme-based model. In their approach, the transliteration equivalents are identified by matching in a target language database based on edit-distance. The authors trained their tool with several names before the transliteration process. Surana and Singh (2008) present a different algorithm that eliminates the training phase. They used fuzzy string matching to account for the lack of training process. Karimi, Turpin and Scholer (2006) split the words into vowels and consonants to achieve transliteration. Their approach focuses on combining most probable combinations of vowels and consonants from source language to target language. A Statistical model for transliteration from English to Arabic words was implemented by Jaleel and Larkey (2003).

3 Methodology

Our algorithm works by converting the Unicode transformation format of source language to its corresponding WX notation taking into account the linguistic knowledge for each language. This WX notation is then converted to the Unicode transformation format of the target language to achieve transliteration. It utilizes the information stored in Unicode transformation format to automatically identify the source language. The target language, however, needs to be specified.

Before we begin with the description of the algorithm, let us first define what Unicode transformation format and WX notation are.

Definition 1: Unicode transformation format (UTF): It is the universal character code standard to represent characters. UTF-8 is an alternative coded representation form for all the characters in Unicode while maintaining compatibility with ASCII (Unicode Standard Version, 2003).

Definition 2: WX-Notation: WX notation is a transliteration scheme to denote a script in Roman script. It defines a standard for the representation of Indian Languages in Roman script. These standards aim at providing a unique representation of Indian Languages in Roman alphabet (Akshar et.al., 1995).

The WX notations for different Indian Languages are similar in their representation (See Table 1). We utilize this property for the development of our algorithm for transliteration.

Language	UTF-8	WX
Hindi	सचिन	Sacina
Bengali	সাচিন	
Telugu	శచీన	
Punjabi	ਸਚਿਨ	
Malayalam	സചින	
Kannada	శచీన	

Table 1: Corresponding UTF and WX for various Indian Languages representing the word “Sachin”

Thus the problem of transliteration can now be divided into sub problems each of which can be addressed by designing converters for converting UTF to WX and WX to UTF for each language.

This method of conversion using an intermediate notation was necessary so as to limit the number of converters required for several languages (Using direct mapping, for 6 languages, we would have required 30 different transliteration tools whereas using the intermediate notation, we just need 6 tools for converting from UTF to WX and another 6 to convert back from WX to UTF thus limiting the number of tools to just 12). Another benefit of this notation is that we can extend it to convert into other languages by simply adding 2 tools that could convert from UTF to WX and vice versa for that language.

3.1 Identifying Source Language

The first step in the transliteration process that we explain in the paper is to identify the source language. The source language of the given text can automatically be detected by analyzing the UTF characters. UTF characters follow a particular order in the representation of characters. All the characters of a particular script are grouped together. Thus we can identify which language/script is presented to the software by analyzing its character codes. UTF-8 characters are variable length. For Indian languages, these characters comprise of three bytes. Thus to detect the script of the UTF-8 characters, we analyzed the three bytes for different languages for some pattern. By comparing the code of second byte, the Indian Languages can be identified (See Table 2).

Language	Code for second byte
Hindi (hin)	164 or 165
Bengali (ben)	166 or 167
Telugu (tel)	176 or 177
Punjabi (pan)	168 or 169
Malayalam (mal)	180 or 181
Kannada (kan)	178 or 179

Table 2: Character Codes for UTF-8 representation of different Indian Languages

3.2 Converting UTF to WX

The next task is to convert the UTF form of language to the corresponding WX notation. This is achieved by using different converters for different languages. These converters are similar in their implementation with a few minor changes for each arising due to its linguistic rules. Firstly, we initialize the character maps which usually represent a many to one mapping from Roman characters to UTF. We then extract characters from the input string one by one. We then push the corresponding WX equivalents of these characters to the output string. We have to keep in mind about maintaining the efficiency of the algorithm so that searching for an element in the map is minimized. For this purpose, we have made a map that corresponds to the indices that we can obtain using UTF characters. Thus we don't need to search the map for UTF characters. Each UTF character has a different code and from that code, we can extract an index that points to its corresponding WX character. This finds the WX equivalent for each UTF character in constant time.

3.3 Converting WX to UTF

Once we obtain the WX notation for the given source text, the next step is to convert the WX notation to UTF of target language. This can be done using a similar mapping of Roman characters to UTF. Again we have to keep in mind about maintaining the efficiency of the algorithm so that searching for an element in the map is minimized. This is done by utilizing the ASCII codes of roman characters that are used to represent WX characters and then building the map as required. Thus WX to UTF conversion for each character is also achieved in constant time.

4 Results

In order to prove our algorithm, we compared the performance of our tool with the results provided on a test set by Linguists having knowledge of both the source as well as target language.

To evaluate our method, we tested our tool on a large corpus having 10k (approx. 240k words) sentences in Hindi. We then transliterated the complete corpus into each of the target languages one by one, results of which are listed in table 3.

Target Language	Accuracy
Hindi	95.8
Bengali	93.2
Telugu	90.0
Punjabi	92.9
Malayalam	85.2
Kannada	87.1

Table 3: Different Indian Languages and corresponding accuracy

The accuracy is based on the phonetic pronunciations of the words in target and source language and this was obtained from Linguistics having knowledge of both the languages.

यहाँ पहुंचना भी मुश्किल नहीं है | राणा सांगा को
श्वाने के लिए सूगलोंगे ते यहाँ कर्षे वार आक्रमण किया जितेंगे
कर्षे वार राणा सांगा घायल हुए | यहाँ का सबसे बआ आकर्त
आकर्त समुद्र का सँगम है | परम्परागत लयमें
पृष्ठेज्यूदा देखने लायक बात राजा रवि
राजू के चित्तपूर्ण।

a) Input Text to transliteration module

यहाँ पहुंचना भी मुश्किल नहीं है | राणा सांगा को हराने के
लिए मुगलों ने यहाँ कई बार आक्रमण किया जिनमें कई बार
राणा सांगा घायल हुए | यहाँ का सबसे बआ आकर्त समुद्र का
सँगम है | परम्परागत में बब्बे ज्यादा देखने लायक बात
राजा रवि वर्मा के चित्र हैं |

b) Output in Hindi

Figure 1: Results of transliteration module

Another important point to note in the transliteration module is its time efficiency. Since it may be used as a part of the complete translation tool, it has to perform its task very rapidly. Keeping this in view during our implementation, we now present the time taken by our tool.

For 100 words written in Devanagari (Hindi), the transliteration into Malayalam using our tool takes less than 0.100 seconds on an Intel Core 2 Duo, 1.8 GHz machine running Fedora.

5 Conclusion

In this paper, we present an algorithm for the efficient transliteration between Indian Languages. We presented a brief overview of UTF and WX notations and then our algorithm that involved transition from UTF to WX of source language and then back to UTF for target language.

6 Future Work

The algorithm presented in the paper is an efficient algorithm for transliteration and would be used in translation between Indian Languages. We are also exploring on how to make the mapping more efficient using automatic learning.

References

- Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1995. *Natural Language Processing : A Paninian Perspective*. Prentice Hall of India.
- Nasreen Abdul Jaleel and Leah S. Larkey. 2003. Statistical transliteration for english-arabic cross language information retrieval. In *Proceedings of the twelfth international conference on Information and knowledge management*, New Orleans, LA, USA.
- Srinivasan C. Janarthanam, Sethuramalingam S, and Udhayakumar Nallasamy. 2008. Named entity transliteration for cross-language information retrieval using compressed word format mapping algorithm. In *Proceedings of 2nd International ACM Workshop on Improving Non-English Web Searching*.
- Sarvnaz Karimi, Andrew Turpin, and Falk Scholer. 2006. English to persian transliteration. In *SPIRE 2006*, pages 255–266.
- Harshit Surana and Anil Kumar Singh. 2008. A more discerning and adaptable multilingual transliteration mechanism for indian languages. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, Hyderabad, India.
- Addison Wesley The Unicode Consortium. 2003. The Unicode Standard, Version 4.0.
- Wikipedia. 2010. Languages of india. http://en.wikipedia.org/wiki/languages_of_india. (accessed: April 21, 2010).

Konstruktionsgrammatische Analyse von Gliederungsstrukturen

Harald Lüngen

Zentrum für Medien und Interaktivität
Justus-Liebig-Universität Gießen
Germany
[{harald.luengen,mariana.hebborn}](mailto:{harald.luengen,mariana.hebborn}@zmi.uni-giessen.de)@zmi.uni-giessen.de

Mariana Hebborn

Zentrum für Medien und Interaktivität
Justus-Liebig-Universität Gießen
Germany
[{harald.luengen,mariana.hebborn}](mailto:{harald.luengen,mariana.hebborn}@zmi.uni-giessen.de)@zmi.uni-giessen.de

Abstract

Dieser Beitrag beschreibt, wie Analysen von Gliederungsfragmenten als Gliederungskonstruktionen im Formalismus der *Sign-Based Construction Grammar* (Sag, 2007) dargestellt werden können.

1 Motivation und Ziele

Wissen in textueller Form tritt uns immer als visuell und hierarchisch gegliederte Einheiten von Text entgegen, was in wissenschaftlichen Texten besonders deutlich wird. Eine Forschungshypothese des laufenden Projekts „Die Ordnung von Wissen in Texten - Textgliederung und Strukturvisualisierung als Quellen natürlicher Wissensontologien“¹ ist, dass die Gliederung wissenschaftlicher Texte wesentliche Teile der Wissensstruktur widerspiegelt, die im Text aufgebaut wird. Die Gliederung einer modernen Dissertation etwa stellt einen Kompromiss dar zwischen den Anforderungen der Textsorte und den methodischen und sachlichen Gegebenheiten des Gegenstandes.

Ziel des Projekts ist es, zu untersuchen und zu beschreiben, wie hierarchisch-visuelle Gliederungssysteme aufgebaut sind, wie Wissensstrukturen in ihnen kodiert werden und wie aus Gliederungen automatisch Ontologien für maschinell unterstützte Navigations-, Archivierungs- oder Suchaufgaben abgeleitet werden können.

2 Korpus

Zur Bearbeitung der Forschungsfragen wurde ein Korpus von 32 digitalen (zumeist PDF) wissenschaftlichen Lehrbüchern aus zwölf Diszipli-

¹gefördert im Rahmen des LOEWE-Schwerpunkts *Kulturtchniken und ihre Medialisierung* an der Justus-Liebig-Universität Gießen

nen zusammengestellt.² Der Textinhalt und die XML-basierte Dokumentstruktur dieser Lehrbücher wurden extrahiert und mittels einer Suite von XSLT-Stylesheets in eine XML-Annotation der Dokumentstruktur nach den TEI-P5-Guidelines überführt. Außerdem wurden die Lehrbuchtexte mittels des Tree Taggers und Chunkers von der Universität Stuttgart (Schmid, 1994) mit morphologischen Analysen und Chunk-Informationen sowie mittels des Machinese Syntax-Parsers von Connexor Oy (Tapanainen und Järvinen, 1997) mit dependenzsyntaktischen Analysen versehen. Alle Annotationsebenen sind in XML realisiert und werden auf der Basis ihrer identischen Primärdaten zu XStandoff-Dokumenten kombiniert. XStandoff-Dokumente repräsentieren Multi-Layer-Annotationen und können wie in (Stührenberg und Jettka, 2009) beschrieben mit Hilfe von XML-Standards und -Werkzeugen verarbeitet werden. Das so aufbereitete Datenmaterial stellt die Grundlage für die weiterführende Analyse der vorliegenden Korpusdaten.

Die Gliederungen der Lehrbücher unseres Korpus sind in der Form automatisch generierter Inhaltsverzeichnisse als XStandoff-Dokumente mit allen verfügbaren Annotationsschichten in der nativen XML-Datenbank eXist gespeichert und werden durch XQuery-Anfragen oder Perl-Programme, die die LibXML-Funktionsbibliothek verwenden, ausgewertet. Diese Korpusinfrastruktur wird genutzt, um Gliederungsfragmentanalysen, wie in (Lüngen und Lobin, 2010) dargestellt, durchzuführen.

²An dieser Stelle möchten wir den Verlagen Facultas, Haupt, Narr/Francke/Attempto, Springer, UTB, Vandenhoeck & Ruprecht, und Wissenschaftliche Buchgesellschaft danken, die uns freundlicherweise digitale Versionen von Lehrbüchern zur Verfügung gestellt haben.

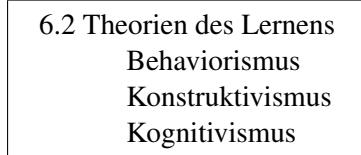


Abbildung 1: Gliederungsfragment

3 Gliederungskonstruktionen

Ausgehend von den Teilaufgaben der Ontologie-induktion „Identifikation von Domänenkonzepten“ und „Identifikation von Relationen zwischen Domänenkonzepten“ (vgl. z.B. Maedche und Staab, 2004) untersuchen wir, wie ontologisch-semantische Relationen zwischen Domänenkonzepten durch Gliederungsstrukturen angezeigt werden. In Abbildung 1 ist ein *Gliederungsfragment* dargestellt, in diesem Fall eine Zwischen-überschrift mit drei ihr unmittelbar untergeordneten Überschriften aus der Gliederungsstruktur des Bandes *Einführung Pädagogik* von (Raithel et al., 2007). Für die Beschreibung der Semantik eines solchen Fragments verwenden wir den Ansatz der *Multilayered Semantic Networks* (MultiNet) von Hermann Helbig (Helbig, 2006). Der MultiNet-Ansatz ist eine umfassende semantische Theorie und bietet ein großes und konsistentes Inventar semantischer Typen, Merkmale, Relationen und Funktionen. Es wurde beispielsweise für die syntaktisch-semantische Analysekomponente eines QA-Systems verwendet (Gloeckner et al., 2007).³

Die MultiNet-Repräsentation der Semantik des Gliederungsfragments in Abbildung 1 ist in Abbildung 2 zu sehen: Die Domänenkonzepte *Behaviorismus*, *Konstruktivismus* und *Kognitivismus* aus den Unterüberschriften stehen jeweils in einer *isA*-Relation (in MultiNet: SUB) zu dem Domänenkonzept *Lerntheorie* aus der übergeordneten Überschrift. Welche linguistischen und strukturellen Eigenschaften führen zu dieser semantischen Interpretation? Es reicht nicht aus, dass die vier linguistischen Ausdrücke in dieser Reihenfolge hintereinander im Text vorkommen, entscheidend ist, dass der Ausdruck *Theorien des Lernens* im Plural steht und dass die drei anderen Ausdrücke jeweils durch eine Unterordnungsrelation im Rahmen einer Gliederungstruktur einge-

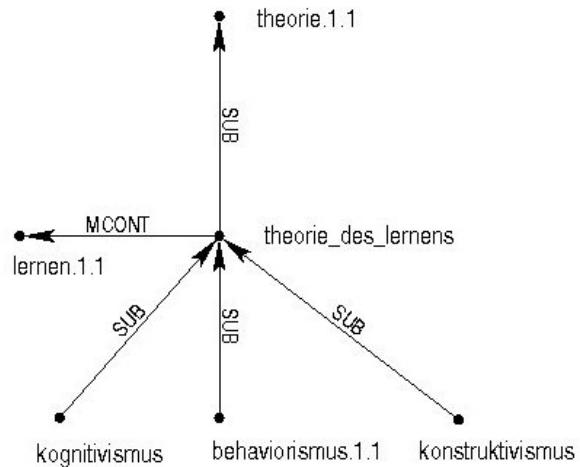


Abbildung 2: MultiNet-Repräsentation der Semantik des Gliederungsfragments



Abbildung 3: Gliederungsschema

führt werden. Informell können wir sagen: Liegt ein *Gliederungsschema* wie in Abbildung 3 vor, so kann ein MultiNet-Schema wie in Abbildung 5 oder durch Mehrfachanwendung ein MultiNet-Schema wie in Abbildung 5 abgeleitet werden, falls folgende Bedingungen gelten:

1. SELF enthält den Term T_1 , der auf das Domänenkonzept B referiert.
2. CHILD enthält den Term T_2 , der auf das Domänenkonzept A referiert.
3. T_1 ist ein Nomen oder eine Nominalphrase im Plural.
4. T_2 ist ein Nomen oder eine Nominalphrase



Abbildung 4: MultiNet-Schema

³Für den Entwurf von MultiNet-Repräsentation wurde uns freundlicherweise der MWR-Editor von Professor Helbig's Gruppe in Hagen zur Verfügung gestellt.

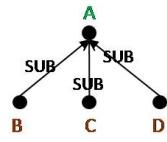


Abbildung 5: MultiNet-Schema

Eine solche Paarung von Gliederungsschema und MultiNet-Schema nennen wir *Gliederungskonstruktion* in Anlehnung an die Konstruktionsgrammatik (Kay, 1995). In der Konstruktionsgrammatik wird die Kombination sprachlicher Ausdrücke durch die Assoziation von Formenschemata mit Bedeutungsschemata beschrieben; Aufgrund ihrem monostratalem Charakter erscheint die Konstruktionsgrammatik besonders geeignet für unsere Beschreibungsaufgabe, nämlich verschiedene sprachliche Ebenen wie Semantik, Syntax und Pragmatik in einer einheitlichen Struktur abzubilden. Eine formalisierte Version der Konstruktionsgrammatik ist die sogenannte *Sign-based Construction Grammar* (SBCG) (Sag, 2007; Michaelis, 2010), welche eine Anwendung des Formalismus der getypeten Attribut-Wert-Matrizen (AWMs) des HPSG-Ansatzes (Pollard und Sag, 1994) auf die Konstruktionsgrammatik darstellt. Abbildung 6 zeigt die oben beschriebene Gliederungskonstruktion als AWM nach dem SBCG-Formalismus.

2-level-cxt-plural-1 ⇒

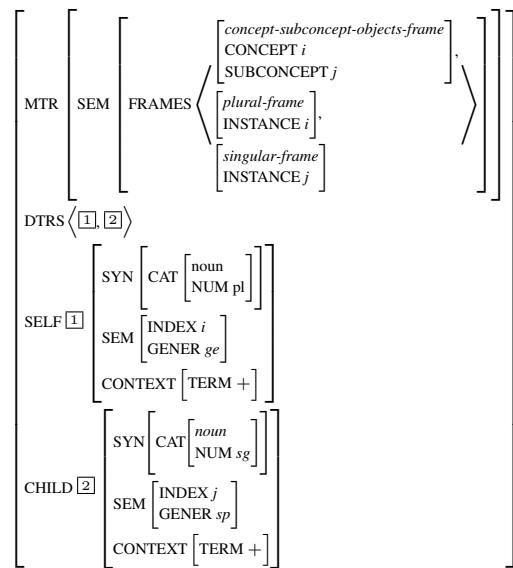


Abbildung 6: Gliederungskonstruktion „PLURAL“ nach SBCG

Abbildung 7 zeigt die Repräsentation des konkreten Konstrukts aus dem obigen Beispiel, welches durch die Konstruktion in Abbildung 6 (und entsprechende Lexikoneinträge) lizenziert wird.

2-level-cxt-plural ist die Typenbezeichnung für die Klasse der Konstrukte, die sich auf zwei Gliederungsebenen beziehen und in deren Fokus sich eine Überschrift mit einem Domänen-Term im Plural befindet. Die Merkmale SELF und CHILD wurden zur Auszeichnung der Unterordnungsrelation in Gliederungsschemata eingeführt; wir verwenden SELF, CHILD, PARENT nach dem Vorbild der XML Path Language (Clark, 1999) um im Gliederungsschema von einer Überschrift, die im Fokus steht (SELF) auf über-, unter- oder nebengeordnete Überschriften Bezug zu nehmen. Im Zusammenhang mit der Koindizierung mit Elementen der DTRS-Liste der Konstruktion erfüllen sie eine ähnliche Funktion wie das HD-DTR-Merkmal in den *headed-constructs* der SBCG (Sag, 2007, S.51). Die bisherige Inventarisierung von Gliederungskonstrukten hat gezeigt, dass sie keine *headed structures* sind, daher wirkt das Head-Feature Principle in diesen Strukturen nicht, wohl aber das Semantik-Prinzip (*Principle of Compositionality*, cf. (Sag, 2007, S.42)), welches die FRAMES-Liste der DTRS mit den FRAMES-Listen der MTR unifiziert. In den FRAMES verwenden wir als Merkmale die C-Rollen aus dem MultiNet-Ansatz (Helbig, 2006), wie beispielsweise MCONT (für mental content) in Abbildung 7. Unter CONTEXT führen wir das boolesche Merkmal TERM ein, welches den Terminologiestatus eines Ausdrucks anzeigen.

4 Ausblick: Ontologieinduktion aus Gliederungsinformationen

Derzeit werden Gliederungskonstruktionen inventarisiert, im SBCG-Formalismus beschrieben und in TRALE (Penn, 2003; Melnik, 2007) implementiert, um sie auf Konsistenz zu überprüfen. Ein weiteres Ziel des Projekts ist die Implementierung eines Prototyps für die Ontologieinduktion aus Gliederungsstrukturen. Bei der Extraktion von Domänenkonzepten und semantischen Relationen zwischen ihnen werden Gliederungskonstruktionen eine ähnliche Rolle spielen wie die lexiko-syntaktischen „Hearst Patterns“ (Hearst, 1992), die bei der Ontologieinduktion aus Fließtext angewendet wurden. Das Poster wird weitere komplexe Beispiele von Gliederungskon-

struktionen und Paaren von Gliederungsfragmenten und MultiNet-Repräsentationen zeigen, die von ihnen lizenziert werden, sowie eine Beschreibung der Verarbeitungspipeline der Ontologieinduktion.

References

- James Clark, Steve DeRose (Hrsg.) (1999). *XML Path Language (XPath). Version 1.0.* <http://www.w3.org/TR/1999/REC-xpath-19991116/>, 3.05.2010.
- Ingo Glöckner, Sven Hartrumpf, Hermann Helbig, Johannes Leveling, Rainer Osswald (2007). Automatic semantic analysis for NLP applications. In: *Zeitschrift für Sprachwissenschaft*, Jg. 26, H. 2, S. 241-266.
- Hermann Helbig (2006). *Knowledge Representation and the Semantics of Natural Language*. Series Cognitive Technologies. Springer: Berlin.
- Paul Kay (1995). Construction Grammar. In: Verschueren, Jef; Östman, Jan-Ola Blommaert Jan (Hg.): *Handbook of Pragmatics. Manual*. Amsterdam: John Benjamins, S. 171 - 177.
- Harald Lünen, Henning Lobin (Erscheint 2010). Extracting semantic relations from tables of contents Erscheint in: *{Proceedings of Digital Humanities 2010}*.
- Marti A. Hearst (1992). Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th International Conference on Computational Linguistics*.
- Alexander Maedche, Steffen Staab (2004). Ontology Learning. In: *Handbook on Ontologies*. Springer: Berlin.
- Nurit Melnik (2007). From "Hand-Written" to Computationally Implemented HPSG. *Research on Language and Computation*. Volume 5, Number 2.
- Laura A. Michaelis (im Erscheinen). Sign-Based Construction Grammar. Erscheint in: *The Oxford Handbook of Linguistic Analysis*. Oxford University Press: Oxford.
- Stefan Müller (2007). *Head-driven Phrase Structure Grammar: Eine Einführung*. Stauffenburg Einführungen Nr. 17. Tübingen: Stauffenburg.
- Stefan Müller (erscheint 2010). *Grammatiktheorie: Von der Transformationsgrammatik zur beschränkungsbasierten Grammatik*. Stauffenburg Einführungen. Tübingen: Stauffenburg
- Gerald Penn, Detmar Meurers, Kordula De Kuthy, Mohammad Haji-Abdolhosseini, Vanessa Metcalf, Stefan Müller, Holger Wunsch (2003). *Träle Milca Environment v. 2.5.0. User's Manual*. <http://utkl.ff.cuni.cz/rosen/public/trale-manual.pdf>, 3.05.2010.
- Carl Pollard, Ivan A. Sag (1994). *Head-driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. University of Chicago Press, Chicago, IL.
- Jürgen Raithel, Bernd Dollinger, Georg Hörmann (2007). *Einführung in die Pädagogik. Begriff - Strömungen - Klassiker - Fachrichtungen*. 2. Aufl. Wiesbaden: Springer: VS Verlag für Sozialwissenschaften.
- Ivan A. Sag (2007). *Sign-based Construction Grammar. An Informal Synopsis*. Unpublished Manuscript, <http://lingo.stanford.edu/sag/papers/theo-syno.pdf>, 3.05.2010.
- Helmut Schmid (1994). Probabilistic Part-of-Speech Tagging using Decision Trees. In: *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK, S. 44-49.
- Maik Stuehrenberg, Daniel Jettka (2009). A toolkit for multi-dimensional markup: The development of SGF to XStandoff. In: *Proceedings of Balisage: The Markup Conference 2009* (Balisage Series on Markup Technologies, 3).
- Pasi, Tapanainen, Timo Järvinen (1997). A non-projective dependency parser. In: *Proceedings of the fifth conference on Applied natural language processing*. Washington D.C.

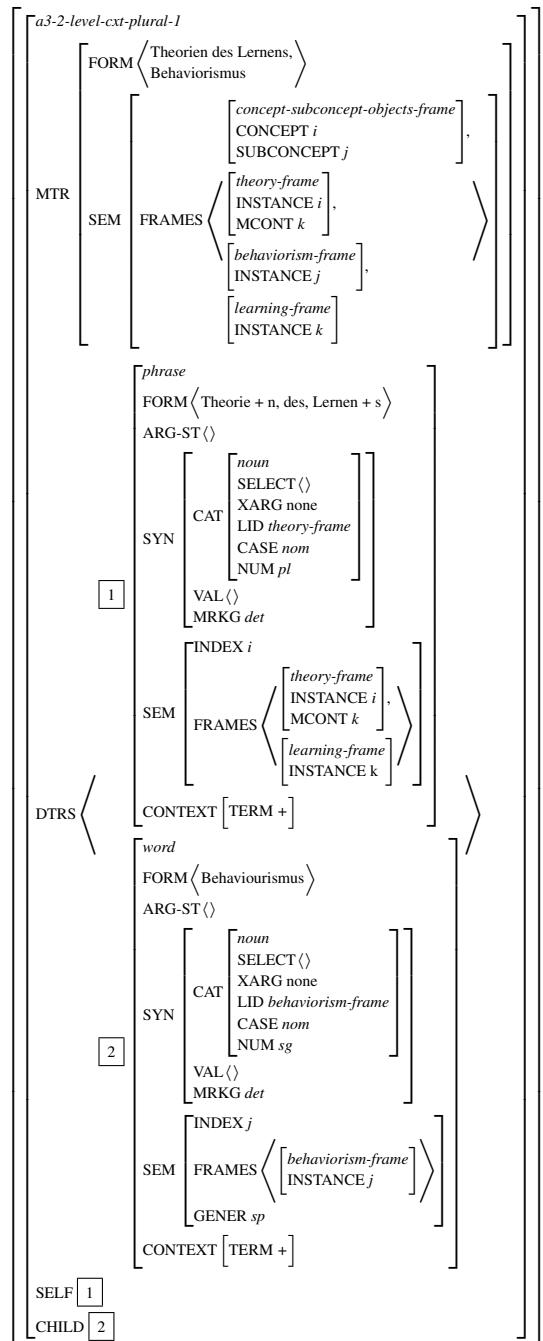


Abbildung 7: Gliederungskonstrukt

Productivity of NPN sequences in German, English, French, and Spanish

Claudia Roch

Katja Keßelmeier
Sprachwissenschaftliches Institut
Ruhr-Universität Bochum
Germany

Antje Müller

{roch, kesselmeier, mueller}@linguistics.rub.de

Abstract

NPN sequences like *step by step* or *year after year* combine regular and irregular aspects of language. While the pattern seems to be fully productive for at least some prepositions, it is highly restricted for others, so that the question of licensing conditions arises. Following Jackendoff (2008) we consider the semantics of the preposition as well as of the noun to play a crucial role here. While some work has been done on NPNs considering examples more or less intuitively, we tackle the question of the productivity of these constructions with the help of corpus-based, statistical methods. We do not only examine NPNs in English but use the EUROPARL corpus for a cross-linguistic comparison of the productivity and the semantics of the NPNs in four languages.

1 Introduction

NPN sequences are combinations of two identical¹ nouns with a preposition in between. The construction includes both idiomatic instances, e.g. *word for word* ('literally') and also more or less regular patterns, e.g. *month after month* (with the analogous forms *day after day*, *book after book*, etc.).

NPN sequences, also known as "binomials", belong to the field of phraseology, which treats irregular aspects of language. Jackendoff, examining English NPN constructions, regards them as "a prime example of what Culicover (1999) calls a 'syntactic nut' - an entrenched noncanonical structure" (Jackendoff, 2008).

Nevertheless he assumes that the construction is fully productive with five English prepositions (*after*, *by*, *for*, *to*, and *upon* (with its variant *on*)), and regards the construction as a whole as semiprimitive. Jackendoff grounds his division into productive and semiprimitive rules on a definition which

says that with a productive rule, the open variable (which is the noun in our case) can be filled freely, while for a semiprimitive rule it has to be learned which cases are acceptable. In his assumptions about productivity, he relies on citations found in the literature and speakers' intuitive judgements.

We study a similar question, namely for which prepositions the NPN construction is productive, using statistical methods. Initially, we tested which prepositions of the entire set of simple prepositions in a language can be found in the construction. In a second step, we determined which of the prepositions are used productively and compared them to each other. An important issue was to detect the determining factors for the regular patterns. Apparently, on the one hand, they depend a lot on the preposition used in the NPN sequence and on its semantics. Possibly, on the other hand, the semantics of the noun plays a crucial role if it turns out that all the nouns in the NPN sequences belong to one semantic class. Therefore, we will include a short section describing the semantics of NPN sequences.

To approach these questions, we chose to do a corpus study using the EUROPARL corpus, which is an aligned parallel corpus consisting of debates held in the European Parliament. It comprises eleven languages, four of which are examined, namely German, English, French, and Spanish.

We use these languages² for a contrastive analysis of NPN sequences. The advantage of a contrastive analysis is that existing productive semantic patterns can be detected cross-linguistically.

The data gained by querying the EUROPARL corpus are fed into the R software³ for statistical computing. As a productivity measure, we use the typical LNRE models (Large Number of Rare

¹NPN sequences with different nouns will be excluded from our investigation.

²As Jackendoff notes, NPN constructions can be found in a multitude of other languages as well, for example in Dutch, Polish, Russian, Japanese or Mandarin.

³<http://www.r-project.org/>

Events) that are implemented in the zipfR package (Baroni & Evert, 2006).

2 Querying the EUROPARL corpus

The EUROPARL corpus contains roughly 40 million tokens per language and is freely available as a part of the OPUS collection (cf. Tiedemann & Nygaard, 2004). It was preprocessed and sentence-aligned by Koehn (2005) and POS-tagged and lemmatized within the IMS Open Corpus Workbench project.⁴

We searched for expressions consisting of two identical nouns joined by a particular preposition, testing the occurrences of all simple prepositions for the different languages one by one. We determined the productivity of the construction by means of vocabulary growth curves. For the calculation of the curves, the total corpus size was reduced by half ten times and each slice of the corpus was treated separately.

The resulting NPN sequences had to be checked manually because such a revision yields more reliable results when it comes to computing productivity. Besides we wanted to exclude some of the sequences. There are some restrictions formulated for the English language by Jackendoff that we can take over: There are no mass nouns or nouns in the plural form allowed (with exceptions), prenominal adjectives are possible, but determiners and post-nominal complements are not permitted in the NPN construction. But in contrast to Jackendoff, NPN constructions with non-identical nouns and NPN sequences proceeded by another preposition (e.g. *from N to N*) are not considered. Proper names will also be excluded.

3 Some initial results

After having clarified the procedure of the extraction and delimited the constructions that will be included in the study, we can take a first glimpse at the total occurrences of NPNs in the corpus.

The total occurrences of the prepositions in NPN sequences across languages can be found in Table 1. There is only a small subset of simple prepositions for each language that can be found in the construction: There are eight prepositions for German (*an, auf, für, gegen, in, nach, über, um*) and English (*after, against, by, for, in, on, to, upon*), and 6 prepositions for French (*à, après, contre, par,*

pour, sur) and Spanish respectively (*a, con, contra, por, sobre, tras*).

p (G)	f	p (En)	f	p (Fr)	f	p (Sp)	f
an	102	after	395	à	348	a	593
auf	5	against	6	après	369	con	100
für	924	by	665	contre	2	contra	2
gegen	6	for	53	par	624	por	474
in	235	in	554	pour	101	sobre	15
nach	5	on	36	sur	11	tras	428
über	11	to	211				
um	106	upon	38				
8	1394	8	1959	6	1455	6	1612

Table 1: Total occurrences of NPN sequences for prepositions in German, English, French, and Spanish

Comparing the total occurrences, the observed languages differ somewhat in their extent of using the NPN construction. The greatest number of NPN sequences can be found in English (1959), followed by Spanish (1612) and French (1455) and finally German with the least number of combinations (1394). Due to the fact that we use a parallel corpus, we can assume that we have a broadly similar text basis for each language. However, the translations in the corpus do not correspond exactly. Often, we find examples of a typical NPN sequence in one language (e.g. German *Schritt für Schritt* 'step by step') which are translated into another language by an adverb (e.g. *gradually*) or by a completely different construction (e.g. *one step after the other*). Sometimes the other language simply does not use an NPN sequence or pieces of text are not translated into the other languages or the alignment does not work correctly in a section.

4 Semantics of the NPN construction

The semantics of the prepositions and the respective nouns play an important role for the semantics of the whole NPN sequence. Jackendoff distinguishes five semantic categories for the NPN constructions, namely 'succession' (in a 'local', 'temporal', and 'local-vertical' sense), 'juxtaposition', 'matching' with the subcategories 'exchange' and 'comparison', and 'large quantity', excluding the interpretation 'transition' that only applies to non-identical nouns.

These categories suggested by Jackendoff for English can also be assigned to the other languages examined, although it must be clear that every language shows idiosyncrasies.

⁴<http://cwb.sourceforge.net/>

	succession	juxtaposition	matching	large quantity
subtypes	local, local-vertical, temporal	-	exchange, comparison, adversative	-
Jackendoff	after, by, upon/on, to	to	for	upon/on
English	after, by, upon/on, to	to, in	against, for	upon
German	auf, für, nach, um	an, in	gegen, um	über
French	à, après, par, sur	à	contre, pour	sur
Spanish	a, por, sobre, tras	a, con	contra, por	sobre

Table 2: Semantic categories for NPN sequences

A semantic classification of the NPN subconstructions for the four languages based on the categories by Jackendoff can be found in Table 2.

The category 'adversative' has been added as a subcategory of 'matching'. It is relevant for the NPN sequences with the preposition *against* or *gegen* (Germ.) respectively, *contre* (Fr.) and *contra* (Sp.) in the other languages.

Apart from the classification based on the different prepositions and their associated meanings, we analyzed the nouns in the NPN sequences. First of all, there are body part expressions and concrete local nouns in the category 'juxtaposition' (cf. Engl. *shoulder to shoulder*, *eye to eye*, *hand in hand*, *arm in arm*, etc.) which can be found across all four languages.

Another important group of nouns contains temporal expressions that occurred in the semantic category 'succession' (e.g. for Engl. *day by day*, *year by year*, *month by month*, *year after year*, *day after day*, *month after month*, *year upon year*, *decade upon decade*). It suggests itself that some kind of 'temporal succession' can be expressed within this category, but the semantic paradigm of nouns is not restricted to temporal expressions here, but open to all kinds of nouns.

This is an interesting observation with regard to the productivity of the construction. In the next section, we present the statistical methods we use to compute the productivity of the NPNs and compare the different prepositions in the subconstructions with each other.

5 Productivity of the NPN construction

Reasonable indicators for productivity are vocabulary growth curves, from which one can glean how the vocabulary will develop on the basis of expected values. Vocabulary growth curves are often applied to compare the productivity of different morphological processes, but will be used here to

compare the productivity of the different subconstructions of NPNs. As the subconstructions differ in vocabulary size, it is necessary to extrapolate the curves to make them comparable.

For extrapolation, models are needed that predict the development of the vocabulary. When dealing with lexical statistic computations, a great number of infrequent occurrences are involved and so the typical LNRE models are required, taking into account these distributions. The statistical computation of these models was done by means of the zipfR package (Baroni & Evert, 2006). Three typical models are implemented there: the Zipf-Mandelbrot model (ZM), the finite Zipf-Mandelbrot model (fZM) and the Generalized Inverse Gauss-Poisson model (GIGP).

The goodness-of-fit was calculated for the three models with every preposition⁵ in an NPN. For each case the best model was selected (the overall best value was reached with a Zipf-Mandelbrot model and the English preposition *on* ($\chi^2 = 0.9123977$, $df = 3$, $p = 0.822435$).⁶

The vocabulary growth curves computed by using the LNRE models and the empirical curves for the NPN sequences for the four languages can be seen in Figure 1. What is principally assumed is that flattening curves stand for unproductive rules, while ascending curves indicate that a process is productive. The vocabulary growth curves indicate that the NPN constructions with the prepositions *um*, *für* (Germ.), *after*, *by*, *for* (Engl.), *après*, *par* (Fr.), and *por*, *tras* (Sp.) are productive. The results for the English NPN constructions agree with the findings of Jackendoff.

⁵Sometimes, these models could not be computed for a preposition because there were too few occurrences in the corpus (cf. Table 1). These prepositions are not included in Figure 1.

⁶The value for χ^2 should be as low as possible and the p-value preferably high, at least > 0.05 .

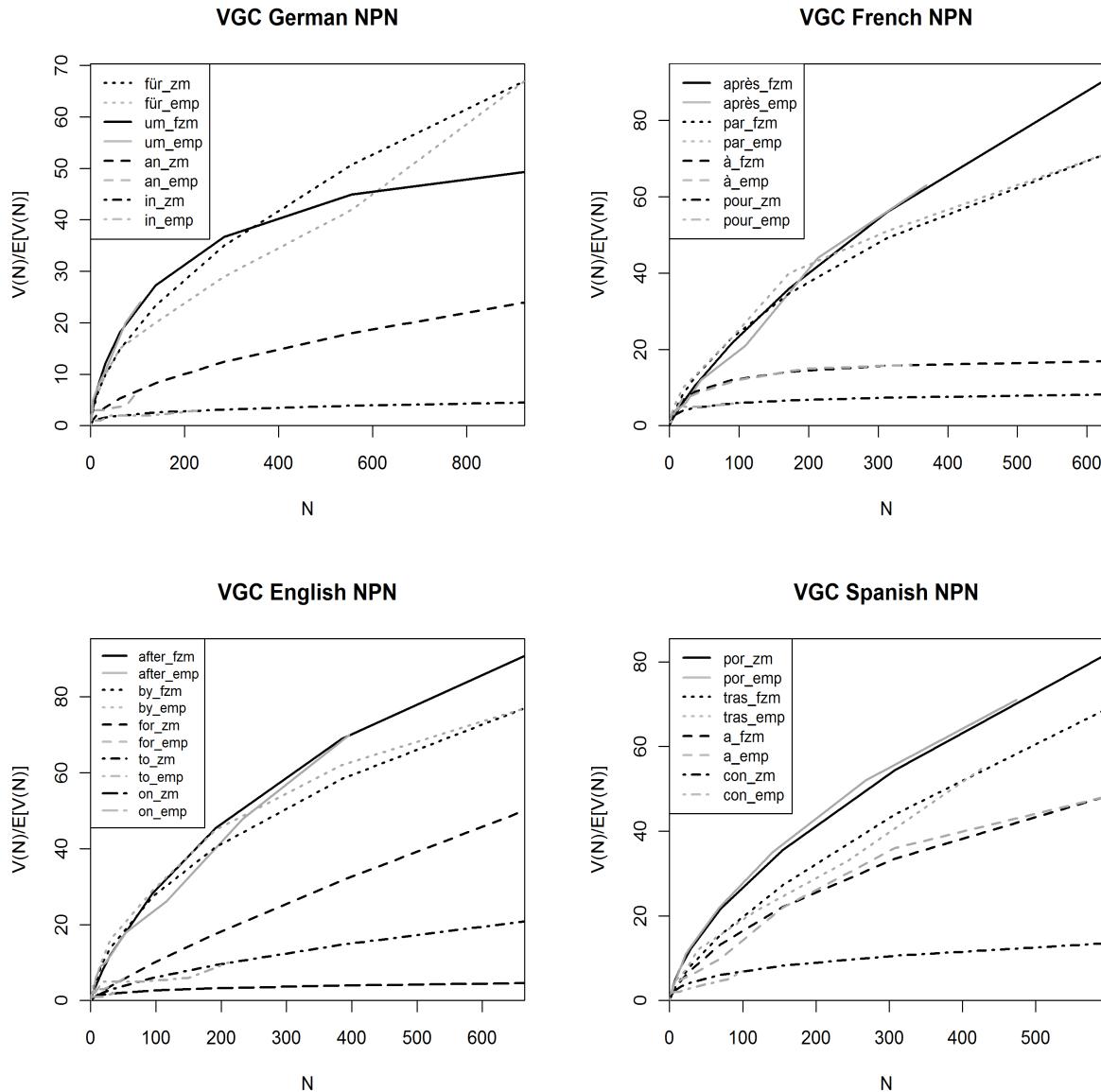


Figure 1: Vocabulary growth curves (empirical and ZM/fZM) for German, English, French and Spanish.

The productive NPN subconstructions are all used to express 'succession', which is clearly an important factor and a cross-linguistic pattern. For the other prepositions, one might suppose that these NPN sequences are more or less unproductive. What must be kept in mind is that a curve may seem productive, but that possibly one has not seen enough data yet.

6 Conclusion

We have shown that the productivity of NPN sequences can be compared by computing the vocabulary growth curves with statistical models. Especially the NPNs with a 'succession' sense seem to

be productive, even across languages, but a more fine-grained semantic classification would be desirable.

As a next step we are planning to repeat the study on a larger German corpus in order to confirm our observations and maybe get even more accurate statistical models for computing the productivity.

Acknowledgments

We would like to thank Jan Strunk and Stefanie Dipper for their helpful comments and Tibor Kiss for his help with R.

References

- Harald Baayen. 2001. *Word frequency distributions*. Kluwer, Dordrecht.
- Marco Baroni and Stefan Evert. 2006. The zipfr package for lexical statistics: A tutorial introduction. URL, [03/05/10]: <http://zipfr.r-forge.r-project.org/materials/zipfr-tutorial.pdf>.
- Peter W. Culicover. 1999. *Syntactic nuts: hard cases, syntactic theory, and language acquisition*. Oxford University Press, Oxford.
- Stefan Evert. 2004. A simple lnre model for random character sequences. In *Proceedings of the 7èmes Journées Internationales d'Analyse Statistique des Données Textuelles*, pages 411–422.
- Ray Jackendoff. 2008. Construction after construction and its theoretical challenges. *Language*, 84(1).
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.
- Jörg Tiedemann and Lars Nygaard. 2004. The opus corpus - parallel & free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.

Automatically Determining the Semantic Gradation of German Adjectives

Peter F. Schulam

Department of Computer Science
Princeton University
United States
pschulam@princeton.edu

Christiane Fellbaum

Department of Computer Science
Princeton University
United States
fellbaum@princeton.edu

Abstract

The semantics of gradable adjectives, in particular their order with respect to one another on a given scale, are not sufficiently represented in lexical resources such as wordnets. Thus, it is not clear whether *superb* expresses the quality of “goodness” equally, more, or less strongly than *excellent* or *great*. Sheinman and Tokunaga (2009) determine the relative gradation of English adjectives by applying lexical-semantic patterns that hold between members of pairs of similar descriptive adjectives to corpus searches. The patterns identify one member of such pairs as the one that expresses a stronger, or more intense, degree of the property denoted by the scale than the other member. By iteratively applying these patterns to a set of adjectives, Sheinman and Tokunaga (2009) arrive at a uniform score for each adjective that allows them to place it at an appropriate point on a scale. We extend the AdjScales method (Sheinman and Tokunaga 2009) to some frequent and salient German adjectives to test its crosslingual robustness. Our work has consequences for automatic text understanding and generation, lexicography and language pedagogy.

1 Introduction

Adjectives remain a relatively ill-understood category. This is reflected in their representation in language resources, which typically lack explicit indications of the degree or intensity with which adjectives express a common property. Thus *Roget’s 21st Century Thesaurus* identifies both *acceptable* and *superb* as synonyms of *good*¹, but native speakers easily agree that the sentences *Her work was acceptable/good/superb* express differ-

ent meanings. Automatic text understanding and generation systems must be able to differentiate among such adjectives as well.

1.1 Adjectives in WordNet

The most widely used lexical resource, WordNet, organizes English adjectives into “dumbbell” structures, consisting of two polar adjectives and a number of adjectives that are semantically similar to one of each of the poles (Gross et al. 1989; Gross and Miller 1990). Polar adjective pairs such as *long* and *short*, called “direct” antonyms by Gross and Miller (1990), label two salient opposed points on a scale such as “length” (Bierwisch 1987). *Extended* and *abbreviated* are “semantically similar” to *long* and *short*, respectively. These adjectives are called “indirect” antonyms of the polar adjectives. Figure 1 shows an example of a WordNet “dumbbell”.

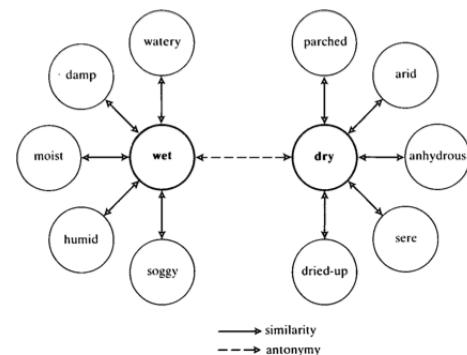


Figure 1: An example of a WordNet “dumbbell structure” (Gross and Miller 1990)

A major shortcoming of WordNet’s treatment of adjectives is that it does not distinguish among the strength of the similar adjectives: what is the degree to which they express the property of the

¹<http://www.thesaurus.com/browse/good>

scale? I.e., what is their relative position with respect to the polar adjectives and one another?

GermaNet abandons the use of WordNet's dumbbell structure and adopts a hierarchical structure akin to that which is used for organizing nouns and verbs (Hamp and Feldweg 1997). The root nodes, like *temperaturspezifisch* ("temperature-specific") refer to the shared property and, in some cases, to the domain of the adjectives in the tree. These terms are the direct superordinates of the polar adjectives. Each polar adjective dominates a subtree of additional adjectives, which are claimed to be increasingly more specific as one descends the tree.

While the Princeton WordNet does not distinguish between the relative intensities of similar adjectives, it encodes a coarse scale by focusing on antonymous adjectives. GermaNet, however, completely abandons a scalar approach in adopting a hierarchical structure. Schulam (2009) questions this move away from bipolarity by showing the strong co-occurrence patterns of German bipolar adjectives — the phenomenon that underpins the Princeton WordNet organization. Furthermore, rejecting a bipolar representation fails to account for the scalar properties of many adjectives.

2 Gradability

Our goal is to assign gradable adjectives to their relative positions on a scale. Kennedy (1999) points out that "a defining characteristic of gradable adjectives is that there is some gradient property associated with their meaning with respect to which objects in their domains can be ordered." A necessary characteristic of gradable adjectives is that they "ascribe to their head nouns values of (typically) bipolar attributes and consequently are organized in terms of binary oppositions" (Gross and Miller 1990). The bipolar adjectives, as well as their similar adjectives, naturally allow themselves to be placed on unidimensional scales, with each pole of the dimension corresponding to one of the two attributes.

2.1 Property Scales

Languages tend to provide words for referring to intermediate values between opposing semantic poles. Consider the phrase "The bread is warm". If we drew a straight line through the lexicalized "points" associated with the words *cold*, *lukewarm*, and *hot*, the word *warm* might lie on the line somewhere between *lukewarm* and *hot*.

Lexical resources indicate that speaker introspection provides only fuzzy judgments about where gradable adjectives fall on a scale. Nevertheless, language provides for ways to detect these subtle gradations via lexical-semantic patterns. Our goal is to empirically identify such patterns for German adjectives and to propose a method for correctly placing them on their respective scales.

3 AdjScales

Sheinman and Tokunaga (2009) propose AdjScales, a method that accepts a group of similar adjectives as input and returns as output a unified scale on which the initial adjectives, as well as additional similar ones, are situated in order of increasing strength.

The fundamental preparatory step is the discovery of scaling patterns (Sheinman and Tokunaga 2009). AdjScales uses pattern extraction queries of the form $a * b$. a and b are seed words and $*$ is a wildcard. Sheinman and Tokunaga perform such queries using a yahoo search engine API that allows for searches and the collection of "snippets", small text excerpts for each result returned from a search engine. These snippets can then be compiled into a database of excerpts containing patterns of interest.

AdjScales selects seed words in a supervised manner using two seeds $seed_1$ and $seed_2$ such that $seed_2$ is *stronger-than* $seed_1$. Sheinman and Tokunaga selected 10 seed word pairs from Gross and Miller (1990) that intuitively display a clear gradation along the same scale.

After successfully creating a database of snippets, AdjScales extracts patterns of the form $[prefix_p \ x \ infix_p \ y \ postfix_p]$ "where x and y are slots for words or multiword expressions" (Sheinman and Tokunaga 2009). Pattern candidates must be consistent with respect to the order in which all instances display the seed words. "Intense" patterns display the weaker word first and the more intense word second, while "mild" patterns do the opposite. Valid patterns must also be supported by at least three seed pairs, and they must repeat twice in extracted sentences. Finally, the patterns must be supported by seed pairs describing different properties.

Pattern extraction is followed by several steps: input, scale relaxation, extension, scaling, and scales unification.

The input step selects at least two similar adjectives. Scale relaxation divides the input adjective set into two antonymous subsets using WordNet’s dumbbell structure (Sheinman and Tokunaga 2009).

The extension step proposes adjectives belonging to the same scale as the input word, based on members of a WordNet dumbbell structure.

Scaling involves iteratively placing pairs of adjectives from the extended input set into the extracted patterns. Sheinman and Tokunaga employ a weighting algorithm that uses the page hit counts returned from a search engine when searching for the complete phrase formed by the adjective pair and the extracted pattern. They use this algorithm to apply a uniform score representing intensity to each adjective (Sheinman and Tokunaga 2009). This allows for the adjective in each of the subdivided groups to be ordered according to increasing intensity as indicated by their score.

After the two subdivided groups have been independently ordered, the scales are unified. The two independently unified scales are merged at the “mild” ends of the spectrum.

4 AdjScales in German

Adapting the pattern extraction process to German adjectives involved selecting suitable seed words, choosing an accessible and extensive corpus in which we could search for patterns, and selecting patterns from the data returned from the pattern extraction queries.

4.1 Seed Word Selection

From a list of 35 antonymous adjective pairs identified by Deese (1964) we selected five antonym pairs as candidate seeds and translated them into German: *kalt-heiß*, *dunkel-hell*, *schnell-langsam*, *traurig-glücklich*, and *stark-schwach*². The pairs represent a variety of different properties to ensure that our extracted patterns would apply to a broad range of semantically diverse adjectives.

Next we paired each of the members of the five antonymous pairs with another adjective from the same scale that we intuitively judged to be more mild or more intense. For example, for *kalt* (*cold*) we chose the milder adjectives *kühl* (*cool*) to complete the seed pair.

We compiled a list of similar adjectives for each of the members of the five antonymous pairs by

²*cold-hot, dark-bright, fast-slow, sad-happy, strong-weak*

Kalt	Kühl	Heiß	Warm
Dunkel	Düster	Hell	Grell
Schnell	Hastig	Langsam	Schlepend
Traurig	Bitter	Glücklich	Zufrieden
Stark	Stabil	Schwach	XXX

Table 1: Complete list of seed words chosen for this study.

#	Intense Patterns
1	X, fast Y
2	X, nicht jedoch Y
3	X, zwar nicht Y
4	X und oft Y
5	X sogar Y
6	X, aber nicht Y

Table 2: List of discovered intense patterns.

using the graphical user interface GermaNet Explorer³, which allowed us to search the adjective trees in GermaNet. After compiling a list of similar adjectives for each of the members of the pairs, we performed queries using COSMAS II⁴

We searched for sentences containing both a translated Deese adjective and one of the corresponding similar adjectives. After iterating through all similar adjectives for each of the Deese adjectives, we chose the most suitable pairing based on the size and diversity of the results returned. The final seed pairs can be seen in table 1. Table cells filled with “XXX” indicate that no appropriate adjective was discovered.

4.2 Pattern Extraction

To extract patterns, we performed queries in COSMAS II that searched for co-occurrences of the seed pairs within the same sentence regardless of their relative order. We exported the results to text files and processed them using simple python scripts.

We first separated the results for each pair of adjectives into files containing “mild” patterns and files containing “intense” patterns. We then removed all results in which the seed words were connected only by *und*, as this pattern does not indicate the relative strength of the adjectives that it joins. The final list of aggregated patterns is shown in tables 2 and 3.

³<http://www.hytex.tu-dortmund.de/ressourcen.html>

⁴<http://www.ids-mannheim.de/cosmas2>

#	Mild Patterns
7	nicht X, aber Y
8	nicht X, aber doch Y
9	nicht zu X, aber Y genug
10	nicht X, sondern Y

Table 3: List of discovered mild patterns.

4.3 Pattern Testing

To test our extracted patterns we used a python script to submit queries containing a pattern in which the adjective slots were filled with test adjective pairs that were different from those used to initially extract the patterns. A test was considered successful when the search returned more results for intense patterns when the adjective pairs were ordered *mild-intense* and when the search returned more results for mild patterns with the adjective pairs were ordered *intense-mild*.

Intense Patterns		Mild Patterns	
Pattern	Hits	Pattern	Hits
1	6	7	0
1	0	8	0
3	0	9	0
4	0	10	0
5	0		
6	628		

Table 4: Page hits for patterns with input *laut-ohrenbetäubend*

Intense Patterns		Mild Patterns	
Pattern	Hits	Pattern	Hits
1	0	7	3
2	0	8	0
3	0	9	0
4	0	10	0
5	0		
6	0		

Table 5: Page hits for patterns with input *ohrenbetäubend-laut*

Many pairs of test adjectives did not return any hits for both the appropriate and inappropriate ordering for both mild and intense patterns. On the other hand, a number of pairs produced successful results. For example, the pair *laut* (*loud*) and *ohrenbetäubend* (*deafening*) returned successful results. The results of this test can be seen in

tables 4 and 5. Given that some pairs of adjectives produced successful results, we believe that the failure of other test cases should not be ascribed to any shortcomings of the method employed but rather to the limited scope of our study. Further work with a larger number of distinct adjectives is needed.

5 Discussion

We demonstrated the robustness of the AdjScales process proposed for English by Sheinman and Tokunaga (2009) by successfully adapting it to German. While the sample set of adjectives used to extract patterns from our test corpus was relatively small — resulting in the aggregation of patterns that could only be applied within a limited scope — our application of the patterns to the selected adjective pairs yielded results mirroring those of Sheinman and Tokunaga (2009). Some of the patterns that we induced for German were translational equivalents of the English patterns, while others seem specific to German. Further extensions to additional languages are planned.

The broader implications of our study is that the recreation of AdjScales strongly supports the underlying semantic analysis. The lexical organization of scalar adjectives into bipolar adjectives (Bierwisch 1987; Gross and Miller 1990) extends across languages that recognize this lexical category. However, the labeling of values along the scale relative to the two poles may well differ crosslinguistically. While linguists, lexicologists and psychologists have long taken this for granted, we believe our contribution to be both novel and important in that it provides empirical evidence for understanding the meanings of specific adjectives as members of a scale. AdjScales can furthermore provide powerful tools for computational linguists and Natural Language Processing applications as well as supply the foundation for the development of more effective language reference tools.

Acknowledgments

We thank the Institut für Deutsche Sprache for making the COSMA corpus available. Fellbaum's work was supported by U.S. National Science Foundation grant IIS 0705155.

References

- M. Bierwisch. *Grammatische und konzeptuelle Aspekte von Dimensionsadjektiven*. Akademie-Verlag, 1987.
- J. E. Deese. The associative structure of some common English adjectives. *Journal of Verbal Learning & Verbal Behavior*, Vol, 3(5):347–357, 1964.
- D. Gross and G.A. Miller. Adjectives in Wordnet. *International Journal of Lexicography*, 3 (4):265, 1990.
- D. Gross, U. Fischer, and G.A. Miller. Antonymy and the representation of adjectival meanings. *Journal of Memory and Language*, 28(1):92–106, 1989.
- B. Hamp and H. Feldweg. Germanet-a lexical-semantic net for german. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, page 9–15, 1997.
- C. Kennedy. *Projecting the Adjective: The Syntax and Semantics of Gradability and Comparison*. Garland Publishing, 1999.
- P. F. Schulam. Restructuring Adjectives in GermaNet, 2009. Unpublished manuscript, Princeton University.
- V. Sheinman and T. Tokunaga. AdjScales: Visualizing Differences between Adjectives for Language Learners. *IEICE TRANSACTIONS on Information and Systems*, 92(8):1542–1550, 2009.

Analytic and Synthetic Verb Forms in Irish – An Agreement-Based Implementation in LFG

Sebastian Sulger

Department of Linguistics

University of Konstanz

Germany

sebastian.sulger@uni-konstanz.de

Abstract

This paper discusses the phenomenon of analytic and synthetic verb forms in Modern Irish, which results in a widespread system of morphological blocking. I present data from Modern Irish, then briefly discuss two earlier theoretical approaches. I introduce an alternative, agreement-based solution, involving 1) a finite-state morphological analyzer for verb forms implemented using the FST toolset (Beesley and Karttunen, 2003); 2) a computational grammar of Modern Irish couched within Lexical-Functional Grammar (LFG) (Bresnan, 2001) and implemented using the XLE grammar development platform (Crouch et al., 2008).

1 Introduction

In Modern Irish, verbs may appear in two different forms: synthetic and analytic. Across tense and mood paradigms, certain person-number combinations are expressed by markers on the verb, resulting in so-called synthetic verb forms. Other person-number combinations are expressed by personal pronouns which appear independent of the verb.

- (1) Present tense paradigm for *tuig* 'understand':

1P.Sg	<i>tuigim</i>	'I understand'
2P.Sg	<i>tuigeann tú</i>	'you understand'
3P.Sg.M	<i>tuigeann sé</i>	'he understands'
3P.Sg.F	<i>tuigeann sí</i>	'she understands'
1P.Pl	<i>tuigimid</i>	'we understand'
2P.Pl	<i>tuigeann sibh</i>	'you understand'
3P.Pl	<i>tuigeann siad</i>	'they understand'

In this example, the forms of the first person singular and the first person plural are synthetic forms. Person and number information is expressed by the ending on the verb. The other forms are an-

alytic verbs which involve separate personal pronouns to express person and number information.

It has been acknowledged in the literature that the function of the person-number endings on the synthetic forms are identical to the function of the independent personal pronouns (Andrews, 1990; Legate, 1999). Evidence for this comes from two observations. First, the use of an independent personal pronoun is prohibited in conjunction with a synthetic verb form.

- (2) *Tuigim mé an fhadhba.
understand.Pres.1P.Sg I ART problem
'I understand the problem.'

(2) is ungrammatical because the person-number information is realized on the synthetic verb form, blocking the use of a separate personal pronoun.

Second, the use of an analytic verb form is blocked if there is a synthetic verb form realizing the same features as the analytic form combined with a pronoun. Since there is a synthetic verb form available in the paradigm for the features '1st person singular' (*tuigim*), the use of the analytic verb form in conjunction with a personal pronoun is blocked.

- (3) *Tuigeann mé an fhadhba.
understand.Pres I ART problem
'I understand the problem.'

An implementation using a computational grammar is thus faced with two separate tasks: 1) block the redundant use of the independent subject pronoun when combined with a synthetic verb form, as in (2); 2) block the analytic verb form when there is a synthetic verb form available, as in (3).

2 Earlier Approaches

Andrews (1990) presents an LFG approach. The approach crucially depends on the mechanism of

unification. More specifically, he proposes a solution in form of a constraint on lexical insertion, the *Morphological Blocking Principle*. Andrews (1990) defines this principle as a variant of the Elsewhere Condition, modified to control lexical insertion in LFG. The principle is formulated as follows:

Morphological Blocking Principle (MBP):

Suppose the structure S has a pre-terminal node P occupied by a lexical item l_1 , and there is another lexical item l_2 such that the f-structure determined by the lexical entry of l_1 properly subsumes that determined by the lexical entry of l_2 , and that of l_2 subsumes the f-structure associated with P in S (the complete structure, after all unifications have been carried out). Then S is blocked.

(Andrews, 1990, p. 519)

For Irish verbs, this principle essentially has the following consequences. When the f-structure of an analytic verb form is unified with the f-structure of an independent pronominal, the lexicon has to be checked to see if there is another verb form that subsumes the resulting unified f-structure (i.e., a form that already contains the pronominal features in its lexicon entry – a synthetic form). If there is such a form, the analytic form is blocked.

An obvious issue with this approach is connected to efficiency. For every verb form occurring in a sentence, the whole lexicon has to be checked for a corresponding synthetic form. While Andrews (1990) claims that a first small implementation by the author computes morphological blocking at a tolerable rate, it remains questionable whether this approach is adequate for larger-scale grammars.

Legate (1999) proposes a treatment of morphological blocking based on agreement. The analysis is couched within the framework of Distributed Morphology, drawing on insights from McCloskey & Hale (1984). It argues that the affixes found on verbs are truly agreement patterns in Modern Irish. The agreement between the verb and the subject pronoun must be realized via an agreeing affix on the verb (i.e. the synthetic form), since these affixes are more specified than the default affix (i.e. the analytic form). The paper departs from earlier literature in Distributed Morphology in that it requires two changes in the vo-

cabulary insertion mechanism. First, the mechanism must operate top-down instead of bottom-up, as was assumed in previous papers. Second, any morpho-syntactic features realized by a vocabulary item have to be deleted. The paper concludes arguing that the Irish data constitutes an interesting argument for a framework of morphology that applies post-syntactically, based on competition.

Legate (1999) also mentions the paper by Andrews (1990), saying that this lexicalist alternative is problematic as it involves trans-derivational comparison (i.e., the MBP), which is a powerful and costly mechanism. Since Andrews (1990) compares the blocking of analytic forms by synthetic forms to the blocking of expressions like *the day before today* by *yesterday*, Legate (1999) concludes that a mechanism like the MBP eventually restricts wordiness.

To sum up, the paper by Andrews (1990) presents a first LFG account for the Irish data, but fails to provide an efficient implementation of the solution, although the approach is theoretically interesting. Legate (1999) makes convincing arguments for an agreement analysis, but, being a theoretical paper, does not offer an implementation; the paper also has to make changes to the applied theory of Distributed Morphology in crucial places.

3 An Alternative LFG Implementation

In this section, I present an alternative LFG approach to the problem of analytic and synthetic verb forms, drawing on theoretical insights from McCloskey & Hale (1984) and Legate (1999). I agree with their work in assuming an agreement relationship between the verb and the subject pronoun. Instead of a competition-based approach (Legate, 1999), my solution constitutes a lexicalist alternative based on agreement and unification, similar to the approaches by Butt (2007) for Punjabi and Bresnan (2001) for Navajo.

My solution uses agreement equations between the verb and the pronominal as a means to block analytic forms from occurring where synthetic forms are available. The implementation is two-fold: 1) A detailed finite-state morphological analyzer (FSMA) dealing with Irish verbal morphology has been written, listing both analytic and synthetic verb forms with morphosyntactic features; 2) a computational LFG grammar has been implemented which effectively rules out analytic verb

forms in inappropriate contexts using two short agreement templates. The implementation is situated in the frame of the ParGram project (Butt et al., 2002). For a related implementation of Welsh morphology, the reader is referred to Mittendorf and Sadler (2006).

3.1 The Finite-State Morphological Analyzer

The FSMA I implemented lists both analytic and synthetic verb forms in the lexicon. All forms are provided with appropriate morphosyntactic tags. The FSMA was implemented using the FST toolkit (Beesley and Karttunen, 2003). In (4), I give the complete present tense paradigm for *tuig* ‘understand’ and the analysis for each of the verb forms.

(4) Pres. tense paradigm for *tuig* & FST analysis:

1P.Sg	<i>tuigim</i>
	<i>tuig+Verb+Pres+1P+Sg+PronIncl</i>
2P.Sg	<i>tuigeann</i>
	<i>tuig+Verb+Pres+2P+Sg</i>
3P.Sg.M	<i>tuigeann</i>
	<i>tuig+Verb+Pres+3P+Sg</i>
3P.Sg.F	<i>tuigeann</i>
	<i>tuig+Verb+Pres+3P+Sg</i>
1P.Pl	<i>tuigimid</i>
	<i>tuig+Verb+Pres+1P+Pl+PronIncl</i>
2P.Pl	<i>tuigeann</i>
	<i>tuig+Verb+Pres+2P+Pl</i>
3P.Pl	<i>tuigeann</i>
	<i>tuig+Verb+Pres+3P+Pl</i>

Notice two things about this analysis. First, the tag *+PronIncl* is attached to synthetic verb forms. This is to make sure that the subject receives a pronominal analysis and a PRED value – details follow in the next section.

Second, the forms are provided with detailed person and number information, even though the verb form is identical in some cases (i.e., the forms are not marked for person/number). A detailed analysis like this enables the grammar to enforce agreement constraints and effectively rule out analytic forms where synthetic forms are available – again, details follow in the next section.¹

¹One reviewer asks whether it would be possible to use a single non-1st-person feature instead of multiple feature sets for the same verb form. This is a question which largely depends on the application for which the FSMA was developed. While the features might not be strictly necessary as input to the LFG grammar to check for agreement facts, they might become a) important to check for in other places in the LFG grammar; b) important to check for by other applications which might be able to make use of the FSMA. Therefore,

3.2 The Computational LFG Grammar

The grammar, implemented using the XLE grammar development platform (Crouch et al., 2008), makes use of the detailed morphosyntactic information provided by the FSMA. The grammar uses a template to enforce agreement restrictions, thereby ruling out analytic forms where synthetic forms are available.

First, I show how the grammar rules out the combination ‘synthetic verb form + independent subject pronoun’ (see (2) for an example). Recall that synthetic forms are provided with the tag *+PronIncl*. Associated with this tag is the following information in the tag lexicon of the grammar:

(5) Information associated with *+PronIncl*:

PronSFX = (\uparrow SUBJ PRED) = ‘pro’

That is, the tag itself provides the information that the subject is a pronominal. In contrast, the lexicon entry of an independent pronoun is given in (6).

(6) mé PRON * (\uparrow PRED) = ‘pro’

(\uparrow PRON-TYPE) = pers

(\uparrow PERS) = 1

(\uparrow NUM) = sg.

When a pronoun like this occurs in the subject position after a synthetic verb form, the unification fails, since there are multiple PREDs – the one supplied by the synthetic form and the one supplied by the pronoun. Multiple PRED features for a single grammatical function are not allowed by LFG, since PRED features are not subject to unification (Bresnan, 2001; Butt, 2007).²

Second, I turn to the more difficult case: how to prevent analytic forms from occurring when synthetic forms are available (e.g., how to rule out sentences like (3)). Recall the detailed morphosyntactic analysis of verb forms outlined in section 3.1. Again, there is functional information associated with each of the tags in (4); see the entries in (7).

I have decided to keep the tags. A related discussion in connection with the German ParGram grammar is whether one should have morphological case tags for nouns which have the same form in all cases, where it was decided to include an .NGDA tag for such nouns (Butt, personal communication).

²One reviewer asks about how ungrammatical sentences such as **tuigeann an fhadhb.* are handled, where the verb form is not synthetic in nature and there is no subject pronoun. Sentences like these essentially violate the principle of completeness in LFG, stating that predators must be satisfied by arguments with semantic features, i.e. PREDs. The above sentence therefore is ungrammatical since the verbal predictor demands a subject argument PRED, and since there is no subject, cannot be satisfied; see also Bresnan (2001).

- (7) +1P V-PERS_SFX XLE @ (AGR-P 1).
+2P V-PERS_SFX XLE @ (AGR-P 2).
+3P V-PERS_SFX XLE @ (AGR-P 3).
+Sg V-NUM_SFX XLE @ (AGR-N sg).
+Pl V-NUM_SFX XLE @ (AGR-N pl).

These entries call up templates in the grammar, passing values over to them. For example, the entry for the morphological tag +2P calls up the template AGR-P and passes over the value 2; similarly, the entry for +Sg calls up the template AGR-N and passes over the value sg. I provide the templates AGR-P and AGR-N in (8) and (9).

- (8) AGR-P (_P) = (\uparrow SUBJ PERS) = _P.
(9) AGR-N (_N) = (\uparrow SUBJ NUM) = _N.

When the value 2 is passed on to AGR-P, the template tries to assign the value to the PERS attribute of the subject; correspondingly, when the value sg is passed on to AGR-N, the template tries to assign the value to the NUM attribute of the subject. The templates effectively result in the unification of features coming from the verb form and the independent pronoun.

For example, assume that an independent subject pronoun occurs after an analytic verb form, as in (10). Then the person and number information of the two words are matched against each other, using these templates.

- (10) *Tuigeann mé an fhadhb.
understand.Pres I ART problem
'I understand the problem.'

The analysis of (10) involves the lexicon entry of *mé* 'I' as given in (6), which assigns the value 1 to the feature PERS. It also involves the verb form *tuigeann*, which, according to the FST analysis in (4), can be either third person or second person, singular or plural. The unification and hence the parse consequently fail, as the template AGR-P tries to assign either third person or second person to the subject, while the lexicon entry for *mé* tries to assign first person to the subject. Figure 1 shows one of the failed parses where XLE tries to unify first person information with second person information.

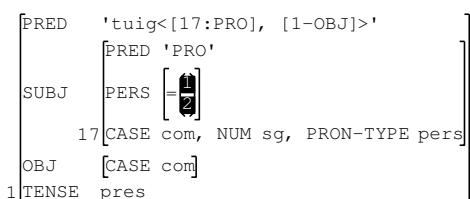


Figure 1: Failed parse of the sentence in (10)

If the information coming from the verb form and the subject pronoun matches, the parse succeeds. In (11), the person and number features of the subject pronoun *sé* agree nicely with the person and number features of the analytic verb form.

- (11) Tuigeann sé an fhadhb.
understand.Pres he ART problem
'He understands the problem.'

The analysis produced by the computational grammar for (11) is shown in Figure 2.³

"tuigeann sé an fhadhb."

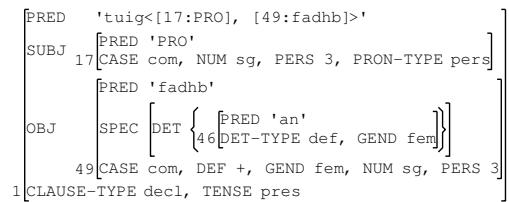


Figure 2: Valid parse of the sentence in (11)

3.3 Evaluation

For evaluation purposes, I manually constructed a testsuite of 30 grammatical and ungrammatical sentences. The implementation currently includes present tense and preterite verb forms in all paradigms and can very easily be extended to include other tenses. The implementation obtains full coverage of the testsuite sentences without any overgeneration.

4 Conclusion

I have presented data from Irish demonstrating the problem of analytic and synthetic verb forms. I have described two earlier approaches; one does not offer an implementation, the other one does offer an implementation, but involves inefficient lexicon checking. I have described my own implementation, which is done using a detailed finite-state morphological analyzer and a computational LFG grammar. The grammar uses efficient templates which rule out non-agreeing verb-pronoun combinations, thereby effectively blocking analytic verb forms where synthetic ones are available.

³One reviewer asks about the speed of the implementation. XLE consists of cutting-edge algorithms for parsing and generation using LFG grammars. It is the basis for the ParGram project, which is developing industrial-strength grammars for a variety of languages. XLE returns the following figures after parsing the sentence in (11):

1 solutions, 0.020 CPU seconds, 0.000MB max mem, 42 subtrees unified

The sentence has 1 solution, it took XLE 0.020 CPU seconds to parse it, and 42 subtrees were unified during the parse.

References

- Avery D. Andrews. 1990. Unification and Morphological Blocking. *Natural Language and Linguistic Theory*, 8:507–557.
- Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications.
- Joan Bresnan. 2001. *Lexical-Functional Syntax*. Blackwell Publishers.
- Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of the COLING-2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.
- Miriam Butt. 2007. The Role of Pronominal Suffixes in Punjabi. In Annie Zaenen, Jane Simpson, Tracy Holloway King, Jane Grimshaw, Joan Maling, and Chris Manning, editors, *Architectures, Rules, and Preferences*. CSLI Publications.
- Dick Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman, 2008. *XLE Documentation*. Palo Alto Research Center.
- Julie Anne Legate. 1999. The Morphosyntax of Irish Agreement. *MIT Working Papers in Linguistics*, 33.
- James McCloskey and Kenneth Hale. 1984. On the Syntax of Person-Number Inflection in Modern Irish. *Natural Language and Linguistic Theory*, 1(4):487–534.
- Ingo Mittendorf and Louisa Sadler. 2006. A Treatment of Welsh Initial Mutation. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG06 Conference*.

A Base-form Lexicon of Content Words for Correct Word Segmentation and Syntactic-Semantic Annotation

Carsten Weber Johannes Handl

Professur für Computerlinguistik

Friedrich-Alexander-Universität Erlangen-Nürnberg

Germany

{cnweber, jshandl}@linguistik.uni-erlangen.de

Abstract

One issue in natural language processing is the interaction between a rule-based computational morphology and a syntactic-semantic analysis system. This is because derivational and compound word forms raise the question of how to deal with ambiguities caused by the rule-based analyser, and how to add additional information like valency to a derivational or compound word form if its valency frames differ from those of its root word.

In this paper we propose a lexicon design addressing both of these issues. We evaluate our design in the context of a large-scale morphological analysis system for German in which the lexicon serves as an interface between morphology and syntax. In doing so, we aim at enriching the well-formed analysis results with additional information so that an adequate syntactic-semantic analysis can be ensured.

1 Introduction

According to di Sciullo and Williams (1987), we consider a word to have both a morphological and a syntactical sense. Accordingly, we distinguish between the *morphologic* and *syntactic word*. The *morphologic word* builds upon morphological principles like inflection, derivation and composition, while the *syntactic word* contains the information which is essential to analyse and interpret a sentence correctly.

In agreement with Trost (1993), we want to separate these two different sets of information. Up to now, our rule-based analysis system handled merely inflection, derivation, and composition, thus providing only the morphological information. In sect. 4, however, we show how to add valency information required for syntactic-

semantic analysis while preserving the strict separation between these two kinds of information.

We chose JSLIM (Handl et al., 2009) as the framework to run our system because it supports feature structures and includes a powerful pre-processor. JSLIM is bound to the SLIM theory of language and builds upon the formalism of LA-grammar (Hausser, 1992). For segmentation, it relies on the allomorph method presented in Hausser (2001). Thus, the lexicon used for morphological analysis is created by several preprocessor steps, as can be seen in fig. 1.

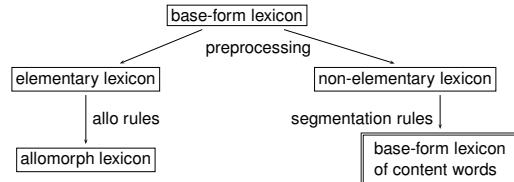


Figure 1: Preprocessor steps

In a first preprocessor step, the entries of the base-form lexicon have to be split into elementary and non-elementary entries.

In a second preprocessor step, we build the allomorph lexicon by executing so-called allo rules on the entries of the elementary lexicon.¹ The result of the execution of the allo rules serves as our lexicon for morphological analysis. The morphological analyser is strictly rule-based and concatenates only valid sequences of allomorphs. At the same time we break the non-elementary entries into their grammatical correct segments and store the output in the *base-form lexicon of content words* (BLCW).² Given the correct segmentation

¹Cf. Handl and Weber (2010).

²The name *base-form lexicon of content words* was first introduced in Hausser (2001).

of all these entries, the BLCW helps to decrease the ambiguity rate during morphological analysis. Furthermore, the BLCW serves as our component for encoding the essential information to be used for syntactic-semantic analysis.

The following sections are intended to give an idea of how our approach works, and to show the advantages of using this supplementary lexicon.

2 Idea

The idea behind using an additional lexicon is to distinguish two kinds of well-formed analysis results. *Well-formed confirmed* means that the input can be decomposed into its allomorph segments, so that the latter can be concatenated correctly by the combination rules. The BLCW contains the segmentation of the input as an entry (key) and the input is therefore an established derivative or compound. *Well-formed unconfirmed* signifies that the input can be decomposed into its allomorph segments, so that the latter can be concatenated correctly by the combination rules. The BLCW does not contain the segmentation of the input as an entry (key) and therefore the input can be a neologism, but is not an established derivative or compound.

Fig. 2 shows the derivation steps of the analysis of the German compound *Haustür* (front-door).³ During morphological analysis the allomorphs which result from segmentation are combined via combination rules. The result are two analyses, one based on the nouns *Haus* (house) and *Tür* (door), the other on the denominal verb *hausen* (to dwell) and the noun *Tür*.⁴ The attribute *seg* stores the base-forms of the allomorphs which result from segmentation, using them as key for lookup in our BLCW at the end of analysis. If the lookup succeeds, the analysis result is marked as well-formed confirmed, otherwise as well-formed unconfirmed. Here, the lookup for the reading as noun-noun compound succeeds, whereas the reading as verb-noun compound fails. Therefore, the noun-noun reading is marked as well-formed confirmed and the verb-noun reading as well-formed unconfirmed.

The distinction between analyses which are well-formed confirmed vs. well-formed un-

³ A detailed introduction to the combination rules of our morphological analyser is given in Handl et al. (2009).

⁴ Lohde (2006, p. 63ff.) points out that noun-noun compounds have a percentage of about 80% in German noun composition, verb-noun compounds in contrary only 5%–10%.

1) rule-based concatenation

LAMMOPH> Haustür

[sur: Haus]	[sur: Tür]	[sur: Haus Tür]
[noun: Haus]	[noun: Tür]	[noun: Haustür]
[cat: (n-g)]	[cat: (f)]	[cat: (f)]
[sem: (n sg)]	[sem: (f sg)]	[sem: (f sg)]
[seg: Haus]	[seg: Tür]	[seg: Haus Tür]

[sur: haus]	[sur: Tür]	[sur: Haus Tür]
[verb: hausen]	[noun: Tür]	[noun: Haustür]
[cat: (v)]	[cat: (f)]	[cat: (f)]
[sem: (imp sg)]	[sem: (f sg)]	[sem: (f sg)]
[seg: hausen]	[seg: Tür]	[seg: hausen Tür]

2) lookup at the base-form lexicon of content words

[noun: Haustür]	-----	matching? -----
-----------------	-------	-----------------

3) new, analyzed results after lookup

LAMMOPH> Haustür

Well-formed confirmed

[sur: Haus Tür]
[noun: Haustür]
[cat: (f)]
[sem: (f sg)]
[seg: Haus Tür]

Yes (1)

Well-formed unconfirmed

[sur: Haus Tür]
[noun: Haustür]
[cat: (f)]
[sem: (f sg)]
[seg: hausen Tür]

Figure 2: Matching mechanism

firmed has several advantages. The first is a decrease in the ambiguity rate: Well-formed unconfirmed results can be suppressed in favor of alternative well-formed confirmed results as can be seen in fig. 2. The statements *Yes (1)* of the analysis output indicates that there is only one single well-formed confirmed reading. The unconfirmed analysis result shown below serves as additional information that the input can also be analysed differently.

Another advantage of the distinction between analyses which are well-formed confirmed and well-formed unconfirmed is in the handling of neologisms. Though the parser does not find a correct segmentation in the BLCW, the input is declared as a well-formed analysis result based on our combination rules. What is more, the label *unconfirmed* hints to the user that the input, though grammatically correct, may not be an established derivational or compound word form. E.g., the German adjective *heizpilzfeindlich* lacks both in well-known traditional dictionaries

as Wahrig (Wahrig, 1997) and online dictionaries like Canoo (Bopp and Pedrazzini, 2009). However, the word occurs in the *Wortwarte*⁵ in an article about outdoor heaters. Since the word *heizpilzfeindlich* is rarely used, it is missing in the BLCW. The latter only comprises established derivatives and compounds. So, the grammar developer can decide, e.g., based on the frequency list of a corpus, whether he wants to add the new word to the lexicon.

Tab. 1 shows the ambiguity rate after analysing a list of types of two well-known German corpora, the NEGRA corpus⁶ and the LIMAS corpus⁷. Our morphology grammar recognizes 95,61% (NEGRA) and 90,83% (LIMAS) of the types from the open word classes of nouns, verbs and adjectives. The ambiguity rates of the accepted types are subdivided with regard to their part of speech. We performed two test runs, one with and one without an integrated BLCW.

Corpus	BLCW	Noun	Verb	Adj
NEGRA	yes	1,48	1,60	1,48
	no	1,79	2,43	1,99
LIMAS	yes	1,41	1,55	1,45
	no	1,80	2,35	1,98

Table 1: Ambiguity rates of the NEGRA and the LIMAS corpus.

The results in tab. 1 reveal a significant reduction of the ambiguity rate for all three open word classes.

It is also interesting to compare the percentage of well-formed confirmed and well-formed unconfirmed analyses. Tab. 2 shows those numbers as well as the respective ambiguity rates.

As can be seen from the test results, there is a discrepancy between the ambiguity rates of the well-formed confirmed and the well-formed unconfirmed analysis results. By improving and completing the BLCW, we also decrease the ambiguity rate.

3 Implementation

As illustrated in fig. 1, we generate the BLCW from the non-elementary lexicon before runtime.

⁵<http://www.wortwarte.de/Archiv/Datum/d090703.html> [July 1st, 2010]. For an introduction cf. Lemnitzer (2007).

⁶Cf. Skut et al. (1997).

⁷Cf. <http://www.korpora.org/Limas/>.

Corpus	Type	Noun	Verb	Adj
NEGRA	confirmed	34,87% (1,28)	65,30% (1,33)	50,69% (1,18)
	unconf.	65,13% (1,72)	34,70% (2,13)	49,31% (1,89)
LIMAS	confirmed	29,90% (1,24)	64,82% (1,31)	48,18% (1,13)
	unconf.	70,10% (1,75)	35,18% (2,15)	51,82% (1,92)

Table 2: Recognition rates of the NEGRA and the LIMAS corpus.

Though the latter already contains established derivatives and compounds, we require supplementary information about

- epenthesis, derivational prefixes and derivational suffixes
- elementary word stems
- and valid allomorph sequences.

According to this set of valid allomorph sequences, we break the entries of the non-elementary lexicon into their segments. Fig. 3 shows three entries of the lexicon, which are broken into more than three allomorphs.

noun: Erholungsurlaub
seg: er holen ung s Urlaub

verb: berechtigen
seg: be Recht ig en

adj: abwechslungslos
seg: ab wechseln ung s los

Figure 3: Entries of the BLCW

The segmentation of the noun *Erholungsurlaub* (holiday) comprises the derivational prefix *er-*, the verb stem *hol* (from *holen*, to fetch sth.), the derivational suffix *-ung*, the epenthesis *-s-* and the noun stem *Urlaub* (holiday). The noun *Erholungsurlaub* is an example for a word form where both derivation (*Erhol-ung*) and composition (*Erholung-s-urlaub*) are active. Besides, an epenthesis occurs between the two nouns.

The verb *berechtigen* (to authorise) may be broken into the derivational prefix *be-*, the noun *Recht* (law), the derivational suffix *-ig* (used to build verbs from nouns), and the inflectional ending *-en*.

The adjective *abwechslungslos* (monotonous) is decomposed into the derivational prefix *ab-*, the

verb stem *wechsl* (from *wechseln*, to change), the epenthesis *-s-* and the derivational suffix *-los*.

These examples show that a precise set of allomorph sequences is essential to decompose complex lexicon entries of the BLCW correctly into their allomorphs.

4 Syntactic-semantic Annotation

Aside from reducing ambiguity by defining the correct segmentation of established derivatives and compounds, the entries of the BLCW can be used to enrich the result of the morphological analysis with syntactic-semantic information.

E.g., during morphological analysis we construct the verb form *gefallen* (to please) by combining the derivational prefix *ge-* with the verb stem *fall* (to fall) and the inflectional ending *-en*.⁸ According to Schumacher et al. (2004), the most relevant valency frames of the verb *fallen* are ⟨nom. compl., verb⟩⁹ and ⟨nom. compl., verb, prep. compl.⟩.¹⁰ The verb *gefallen* has ⟨nom. compl., verb, dat. compl.⟩¹¹ as its most frequent valency frame.

Since the valency frame may change with each combined prefix, it is rather cumbersome to handle the valency frame in its entirety during morphological analysis. We consider the valency information to be a basic, inherent, lexical property which has to be encoded independently from the strictly surface-based concatenation of the allomorphs and thus encode it within the BLCW. By performing a lexical lookup at the end of a morphological analysis we can aggregate the analysis result with this indispensable information for syntactic-semantic analysis.

Fig. 4 shows two entries for the content words *gefallen* and *aufgeben* (to give up sth.) in the BLCW which also comprises valency information. In order to encode different alternative valency frames, JSLIM provides the multicut notation¹², which is also illustrated in fig. 4.

The alternatives of a multicut are enclosed in angle brackets and are in equal measure possible. Hence, phrases like *ich gebe auf* (I give up) and *ich gebe die Hoffnung auf* (I give up the hope) can

⁸Grimm et al. (2001) explain that, originally, the verb *gefallen* was a fortification of the verb *fallen*.

⁹Der Wechselkurs fällt (The exchange rate falls).

¹⁰Der Mann fällt auf sein Gesicht (The man falls on his face).

¹¹Der Mann gefällt mir (The man pleases me).

¹²Cf. Hausser (2001, p. 286).

verb: gefallen	verb: aufgeben
seg: ge fallen	seg: auf geben
cat: ⟨n' d'⟩	cat: <⟨n'⟩(n' a')>

Figure 4: Aggregation of the BLCW with valency information

both be analysed as well-formed confirmed since the accusative complement is encoded as an optional complement. The interaction between the morphological analysis system and the BLCW ensures an adequate handling of two of the basic principles of syntax, namely agreement and valency.

Our approach follows mainly the observation of Ágel (1995) who separates between *valency potential* and *valency realization*. The valency potential, which we define in the BLCW, predetermines the grammatical structures to be realized in the syntax. The valency realization, on the contrary, not only depends on the valency potential of the entries, but also on the rules of our system for a syntactic-semantic analysis. A detailed introduction to syntactic-semantic analysis using LA-grammar is provided by Hausser (2006).

5 Conclusion

We presented an approach which uses an additional lexicon to serve as an interface between morphology and syntax. The BLCW helps to encode the correct segmentations of established derivatives and compounds. Generated automatically before runtime from a non-elementary lexicon, it can be used during morphological analysis for decreasing ambiguity. Though, there is certainly room for improvements regarding segmentation. Further refinements may include a more precise definition of valid allomorph sequences, especially in the case of noun derivatives and compounds. Integrating more established derivatives and compounds could also lead to reduced ambiguity rates. Besides, the BLCW enables the grammar developer to assign syntactic and semantic information of derivational and compound forms in a consistent way. We have shown how valency information can be added to the lexicon to improve syntactic-semantic analysis.

For future research, we plan to annotate a high number of lexical entries with this kind of information and to test them on large-scale corpora.

References

- Vilmos Ágel. 1995. Valenzrealisierung, Grammatik und Valenz. *Zeitschrift für germanistische Linguistik*, 23:2–32.
- Stephan Bopp and Sandro Pedrazzini. 2009. Morphological Analysis Using Linguistically Motivated Decomposition of Unknown Words. In Cerstin Mahlow and Michael Piotrowski, editors, *State of the Art in Computational Morphology. Workshop on Systems and Frameworks for Computational Morphology, SFCM 2009, Zurich, Switzerland, September 4, 2009, Proceedings*, volume 41 of *Communications in Computer and Information Science*, pages 108–117, Zürich. Springer.
- Anna-Maria di Sciullo and Edwin Williams. 1987. *On the Definition of Word*. MIT Press, Cambridge, Massachusetts.
- Jacob Grimm, Helmut Grimm, and Hans-Werner Bartz, editors. 2001. *Deutsches Wörterbuch von Jacob und Wilhelm Grimm im Internet*. DFG-Projekt "DWB auf CD-ROM und im Internet", Univ. Trier, Trier.
- Johannes Handl and Carsten Weber. 2010. A Multilayered Declarative Approach to Cope with Morphotactics and Allomorphy in Derivational Morphology. In *Proceedings of the 7th conference on Language Resources and Evaluation (LREC)*, pages 2253–2256, Valletta, Malta.
- Johannes Handl, Besim Kabashi, Thomas Proisl, and Carsten Weber. 2009. JSLIM - Computational Morphology in the Framework of the SLIM Theory of Language. In Cerstin Mahlow and Michael Piotrowski, editors, *State of the Art in Computational Morphology. Workshop on Systems and Frameworks for Computational Morphology, SFCM 2009, Zurich, Switzerland, September 4, 2009, Proceedings*, volume 41 of *Communications in Computer and Information Science*, pages 10–27, Zürich. Springer.
- Roland Hausser. 1992. Complexity in Left-Associative Grammar. *Theoretical Computer Science*, 106(2):283–308.
- Roland Hausser. 2001. *Foundations of Computational Linguistics. Human-Computer Communication in Natural Language*. Springer, Berlin Heidelberg, second revised and extended edition.
- Roland Hausser. 2006. *A Computational Model of Natural Language Communication*. Springer, Berlin Heidelberg.
- Lothar Lemnitzer. 2007. *Von Aldianer bis Zauselquote. Neue deutsche Wörter. Wo sie herkommen und wofür wir sie brauchen*. Narr, Tübingen.
- Michael Lohde. 2006. *Wortbildung des modernen Deutschen: ein Lehr- und Übungsbuch*. Narr, Tübingen.
- Helmut Schumacher, Jacqueline Kubczak, Renate Schmidt, and Vera de Ruiter. 2004. *VALBU - Valenzwörterbuch deutscher Verben*, volume 31 of *Studien zur deutschen Sprache*. Narr, Tübingen.
- Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An Annotation Scheme for Free Word Order Languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, Washington, DC, USA.
- Harald Trost. 1993. Coping with Derivation in a Morphological Component. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 368–376, Utrecht, The Netherlands. Association for Computational Linguistics.
- Gerhard Wahrig. 1997. *Wörterbuch der deutschen Sprache*. Deutscher Taschenbuch Verlag, München.

Starting a sentence in L2 German – Discourse annotation of a learner corpus

Heike Zinsmeister

Department of Linguistics
University of Konstanz
Konstanz, Germany

heike.zinsmeister@uni-konstanz.de

Margit Breckle

Department of German Philology and Didactics
Vilnius Pedagogical University
Vilnius, Lithuania

margit.breckle@gmx.de

Abstract

Learner corpora consist of texts produced by second language (L2) learners.¹ We present ALeSKo, a learner corpus of Chinese L2 learners of German and discuss the multi-layer annotation of the left sentence periphery – notably the *Vorfeld*.

1 Introduction

Learner corpora consist of texts produced by foreign language (L2) learners. Normally, they are designed as *comparable corpora* which consist of pairs of monolingual corpora selected according to the same set of criteria. In the case of learner corpora, they comprise similar texts in one target language produced by speakers with different L1 backgrounds or with different L2 levels. Furthermore, for reasons of comparison the corpus can contain similar texts by L1 speakers of the target language.

There are two main approaches for investigating the data in a learner corpus (cf. Granger (2008): 267–268): (i) *contrastive interlanguage analysis* (CIA), which assumes that L2 learners use an interim language that differs from the target language in a way that can be observed quantitatively, and (ii) *computer-assisted error analysis*, in which divergences in the L2 texts are identified (and possibly also annotated) based on a target hypothesis.

The current project deals with the creation of the ALeSKo learner corpus², which contains texts from *Chinese L2 learners of German* and is complemented by comparable L1 German texts. Our main interest lies in the expression of *local coherence* – whether the learners acquire the linguistic

means to express a smooth flow from one sentence to the next in German. In the project's current state, we carry out linguistic annotation to create a basis for a CIA of local coherence. Systematic error tagging is not yet performed.³

It is assumed that local coherence is mainly expressed at two levels cross-linguistically (e.g. Reinhart (1980): 168f.; (1982): 19): It is either supported by coreferential entities that play a role in a sequence of sentences (*entity-based coherence*) or it is supported by discourse relations that relate two clauses and also larger parts of the text semantically (*discourse relation-based coherence*). In the current study, we concentrate on entity-based coherence and on the question how it is expressed in the sentence beginnings since both languages – the learners' L1 Chinese as well as their L2 German – do not restrict the initial position in the sentence to a particular grammatical function (i.e. the subject). The position presents itself as an ideal position for linking a sentence to its preceding discourse and establishing local coherence.

Chinese is a *topic-prominent* language. Hence, its general word order and notably its left periphery is strongly determined by information-structural conditions: the topic always comes first which can either be a time or a locative phrase or a familiar referent, for example a referent that is known from the preceding discourse (Li and Thompson (1989): 15, 85f., 94f.). German is a *verb-second* language and provides an initial sentence position (*Vorfeld*), which precedes the finite verb. In contrast to Chinese, German is not strictly topic-prominent even though information

¹For a comprehensive list of learner corpora see www.uclouvain.be/en-cecl-icWorld.html.

²ALeSKo: ling.uni-konstanz.de/pages/home_zinsmeister/alesko.html.

³Multi-layer error tagging is discussed in Lüdeling et al. (2005). For a recent overview of error-tagged corpora see Hana et al. (2010).

structure influences the realisation of the *Vorfeld* (e.g. Molnár (1991); but see e.g. Frey (2004); Speyer (2007) for critical discussions).

Our working hypothesis is that the Chinese learners transfer rules of using the left periphery of a sentence in their L1 Chinese to their L2 German to assure local coherence and hence will show an overuse or an underuse of certain functions in comparison with L1-German speakers.

The rest of the paper presents the ALeSKo corpus and its (entity-based) discourse annotation. We conclude the paper by briefly discussing results from a contrastive interlanguage analysis of entity-based coherence.

2 Related Work

The linguistic annotation of learner corpora is a relatively recent development. The *International Corpus of Learner English* (ICLE)⁴ is the largest project to date. It is responsible for creating a large database of comparable L2-English texts from speakers with a variety of L1s (currently of about 25 different L1s).

The multi-layer annotation of the German error-annotated FALKO corpus⁵ is used as a prototype for the current project's annotation efforts.⁶

Albert et al. (2009) report on error tagging of a learner corpus of French L2 learners of English and a decision model for the best error correction derived from the annotation. The workshop series *Automatic Analysis of Learner Language* (AALL 2008, 2009) brought together various projects of L2-corpus developers and developers of Natural Language Processing applications for foreign language teaching.

The transfer of information structure between two verb-second languages and the filling of the *Vorfeld* is contrastively investigated by Bohnacker and Rosén (2008). However, their analysed data is not published as a reusable annotated corpus.

There have been various research efforts concerning the discourse annotation of L1 corpora. The current project adapts the annotation guidelines for coreference annotation and bridging

⁴ICLE: cecl.fltr.ucl.ac.be/Cecl-Projects/Icle/icle.htm

⁵FALKO: <http://www.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/forschung-en/falko>

⁶The German L1 texts that we report on belong to the FALKO corpus (Falko Essays L1 0.5) and are enriched with additional layers of discourse annotation in the current project.

by MATE (Poesio, 2000), information structure as applied to the Potsdam Commentary Corpus (Götze et al., 2007) and the implicit guidelines of centering annotation from Speyer (2005; 2007).⁷

3 Data

3.1 Collection

The corpus consists of 43 argumentative essays of Chinese L2 learners of German in which they discuss pros and cons of a given subject and state their own opinion. The learners were students at the Konstanz University of Applied Sciences, studying in the program *Business German and Tourism Management*⁸ with a German level of about B2. In addition to the L2 texts, the ALeSKo corpus contains essays by L1 German high school students (aged 16–19) from Berlin, which originally belong to the FALKO corpus. In sum, the Alesko subcorpora include the following texts:

- **wdt07:** 25 L2 texts on the topic *Are holidays an unsuccessful escape from every-day life?* (6,902 tokens, 30–45 min, written exam, no aids)
- **wdt08:** 18 L2 texts on the topic *Does tourism support understanding among nations?* (6,685 tokens, 90 min., dictionary permitted)
- **Falko Essays L1 0.5:** 39 essays on different topics (34,155 tokens, typed in notepad, no internet access, no spell-checker).

The metadata for each individual text provides information about the person's ID, the L1, the year of birth, the gender, the study programme, the foreign language(s), the length of L2 exposure – if applicable – and the essay topic.

3.2 Preprocessing

The hand-written L2 learner texts were manually transcribed. All texts (both L2 and L1) were tokenized, lemmatized and part-of-speech tagged with the TreeTagger (Schmid, 1994). We used EXMARaLDA (Schmidt, 2004) for annotating topological fields in the tagged data. The annotation output of this annotation was converted into

⁷In addition, we annotate discourse relations adapting the guidelines of the Penn Discourse Treebank (Prasad et al., 2007) which is not discussed in this paper.

⁸German: *Wirtschaftssprache Deutsch und Tourismusmanagement* (WDT).

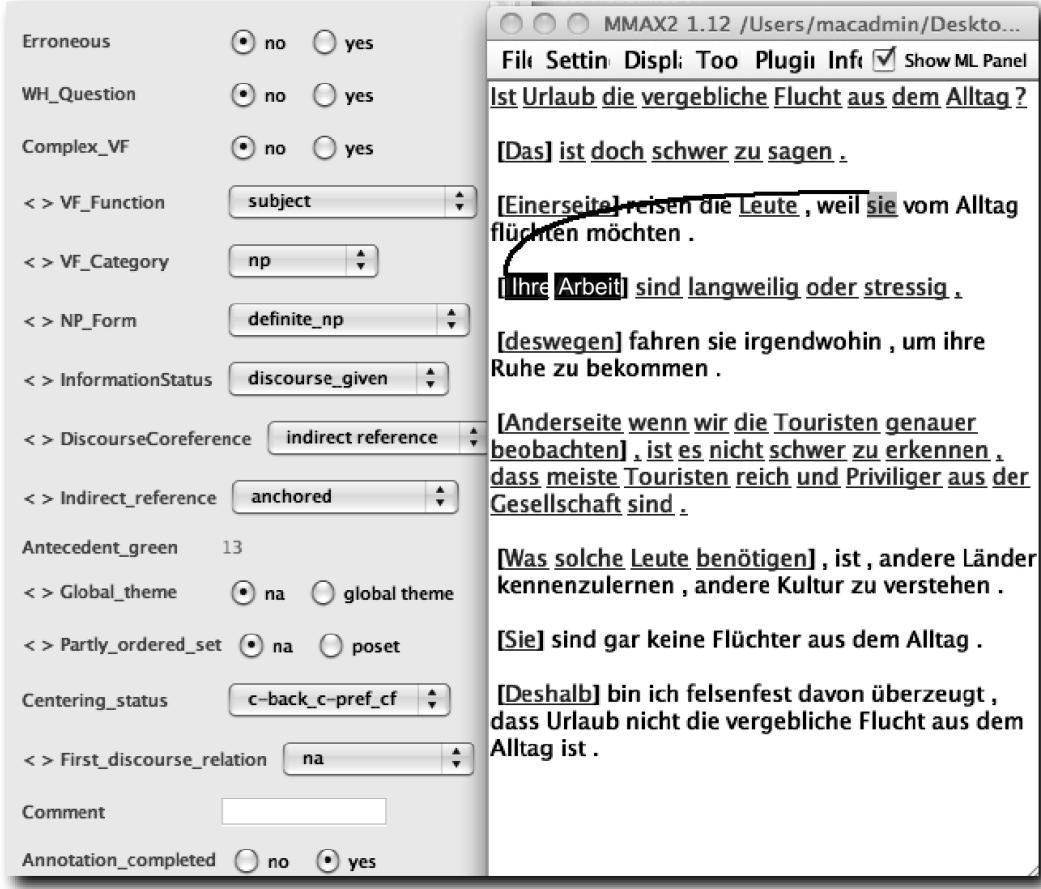


Figure 1: The MMAX2 annotation window of ALeSKo

the MMAX file format and served as the input for the current discourse annotation.

3.3 Annotation

For the investigation of the L2 learners' use of the sentences' left periphery as compared to its use by native speakers, both the L2 texts and the L1 texts are annotated with functional labels. The annotation is performed manually, using the tool MMAX2 (Müller and Strube, 2006), see Figure 1.

Our annotation guidelines define the labels, illustrate them with examples and discuss problematic cases (Breckle and Zinsmeister, 2009). The annotation proceeds in two steps: after a primary annotation by three student annotators or one of the authors, the two authors agree on a gold annotation for each *Vorfeld*.

Table 1 illustrates the relevant layers of annotation with examples.⁹ Figure 1 shows a snapshot of the annotation of example (2) in the annotation tool MMAX2.

⁹The *Vorfeld* constituent is underlined in the English translation.

Information status (*new, deictic, discourse given*, cf. Götze et al. (2007)): The referent of the definite NP *Die Leute, die viele Reise machen* in example (1) is mentioned for the first time and is annotated *new* even though the term *Leute* as such occurs earlier in the text. The referent of *Ihre Arbeit* in (2) is also mentioned for the first time. However, it is a case of indirect reference in which the newly introduced referent is anchored by the possessive pronoun *Ihre* which refers to a discourse-old referent. *Ihre Arbeit* is therefore annotated *discourse_given*.

Partly-ordered set relation (*poset*, cf. Speyer (2005); Prince (1999)): In example (3) *jeden Morgen* ('every morning') and *jeden Abend* ('every evening') form a set of *Tageszeiten* ('times of the day').¹⁰

Centering (*forward-looking center, preferred center, backward-looking center*, cf. Grosz et

¹⁰The poset relation is similar to the concept of contrastive topic (cf. Büring (1999)) which should be taken into account in future revisions of the corpus. Thanks to one of the reviewers for pointed this out to us.

Information status

- (1) [Die Leute, die viele Reise machen,]_{new} haben immer mehr Geld als die, die selten reisen.
‘The people who travel a lot always have more money than those who seldom travel.’
- (2) Einerseite reisen die Leute₁, weil sie₁ vom Alltag flüchten möchten.
[Ihre₁ Arbeit]_{given} sind langweilig oder (...).
‘On the one hand people travel because they want to escape every day life.
Their job is boring or (...)’

Partly-ordered set relation

- (3) [Jeden Morgen]_{element 1} stehen wir auf, um pünktlich zur Arbeit zu sein. (...)
[Jeden Abend]_{element 2} bleiben wir zu Hause, sehen sinnlose Serien im Fernsehn.
‘Every morning, we get up for being at work in time. (...)
Every evening, we stay at home, watch the senseless shows on TV.’

Centering

- (4) Durch Reisen können sie₁ auch andere Kultur und Lebenstile kennenlernen.
[Sie]₁ backward-looking center können auch ihre Kenntnisse durch Reisen erweitern.
‘By travelling, they can become acquainted to other culture and lifestyles.
They can also broaden their knowledge by travelling.’

Internal functions (frame-setting)

- (5) [Heutzutage]_{frame-setting(temporal)} gelangt es in hoher Konjunktur, einen Urlaub zu machen.
‘Nowadays many people go on holidays.’
- (6) [In den Attraktionspunkten]_{frame-setting(local)} werden (...) notwendige Einrichtungen konzentriert angeboten.
‘Necessary facilities are especially offered at the attraction sites.’

Table 1: Examples of discourse annotation in ALeSKo

al. (1995)): In example (4) *Sie* in the second sentence is a *backward-looking center* – the referential expression that corefers with the most salient expression in the previous sentence according to a saliency hierarchy (in comparison to other antecedents): subject is more salient than object(s), object is more salient than other functions.

In addition, **sentence-internal functions** are marked (*frame*: frame-setting topic (Götze et al. (2007): 167f.) and others): Example (5) and (6) present two *frame-setting* elements (temporal and local). They do not contribute to local coherence but they set the frame for the interpretation of the current sentence and are frequently used in the *Vorfeld* in L1 German (cf. Speyer (2007)).

4 Results and Conclusion

We performed a contrastive interlanguage analysis on the basis of the discourse annotation described in section 3. To this end, we compared the relative frequencies of the different functions in the *Vorfelds* of all 43 L2 essays with those in 24 of the Falko L1 essays. With respect to information status (including *poset*) and frame-setting

elements, there is no statistical significant difference between L1 and L2 speakers. However, L2 speakers use the function *backward-looking center* significantly more often in the *Vorfeld* than L1 speakers.¹¹

A more detailed discussion of the analysis is given in Breckle and Zinsmeister (in preparation). Under the (simplifying) assumption that the backward-looking center corresponds to the sentence topic we analyse the observed preference as a transfer effect from the topic-prominent L1 Chinese to the L2 German.

Acknowledgments

Heike Zinsmeister’s research was financed by Europäischer Sozialfonds in Baden-Württemberg

¹¹287 out of 884 (32 %) Vorfelds constituents in the L2 essays function as backward-looking center vs. 207 out of 764 (27 %) in the L1 essays; $\chi^2=5.61$, df=1, $p<0.05$. The conclusion is still valid when the scores are normalised by the lengths of the texts.

References

- Camille Albert, Laurie Buscail, Marie Garnier, Arnaud Rykner and Patric Saint-Dizier. 2009. Annotating language errors in texts: investigating argumentation and decision schemas. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, ACL 2009, 130–133. Singapore, Singapore.
- Ute Bohnacker and Christina Rosén. 2008. The clause-initial position in L2 German declaratives: Transfer of information structure. *Studies of Second Language Acquisition*, 30: 511–538.
- Margit Breckle and Heike Zinsmeister. 2009. Annotationsrichtlinien Funktion des Vorfelds. Manuscript. December 2009. Pedagogical University Vilnius and University of Konstanz.
- Margit Breckle and Heike Zinsmeister. In preparation. A corpus-based contrastive analysis of local coherence in L1 and L2 German. In *Proceedings of the HDLP conference* Frankfurt/Main [a.o.]: Peter Lang.
- Daniel Büring. 1999. Topic. In Peter Bosch and Rob van der Sand (eds.) *Focus – Linguistic Cognitive and Computational Perspectives*, 142–165. Cambridge: Cambridge University Press.
- Werner Frey. 2004. A medial topic position for German. *Linguistische Berichte* 198. 153–190.
- Michael Götze, Cornelia Endriss, Stefan Hinterwimmer, Ines Fiedler, Svetlana Petrova, Anne Schwarz, Stavros Skopeteas, Ruben Stoel and Thomas Weskott. 2007. Information structure. In S. Dipper, M. Götze and S. Skopeteas (eds.) *Information Structure in Cross-Linguistic Corpora: Annotation guidelines for phonology, morphology, syntax, semantics, and information structure* (Working Papers of the SFB 632, Interdisciplinary Studies on Information Structure (ISIS) 7), 94–137.
- Sylvaine Granger. 2008. Learner corpora. In Anke Lüdeling and Merja Kytö (eds.) *Corpus Linguistics. An International Handbook*, 259–275. Berlin / New York: de Gruyter.
- Barbara Grosz, Arvind Joshi and Scott Weinstein. 1995. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21, 203–225.
- Jirka Hana, Alexandr Rosen, Svatava Škodová and Barbora Štindlová. 2010. Error-Tagged Learner Corpus of Czech. In *Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV)*, ACL 2010, 11–19. Uppsala, Sweden.
- Charles N. Li and Sandra A. Thompson. 1989. *Mandarin Chinese: A Functional Reference Grammar*. Berkeley and Los Angeles, CA: University of California Press.
- Anke Lüdeling, Maik Walter, Emil Kroymann and Peter Adolphs. 2005. Multi-level error annotation in learner corpora. In *Proceedings of Corpus Linguistics 2005*, Birmingham, Great Britain.
- Valéria Molnár. 1991. *Das TOPIK im Deutschen und Ungarischen*. Stockholm: Almqvist & Wiksell International.
- Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn and Joybrato Mukherjee (eds.) *Corpus Technology and Language Pedagogy*, 197–214. Frankfurt/Main [a.o.]: Peter Lang.
- Massimo Poesio. 2000. Coreference. In Andreas Mengel et al. (eds.) *MATE Dialogue Annotation Guidelines*, 134–187.
- Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo and Bonnie Webber. 2007. The Penn Discourse Treebank 2.0 Annotation Manual. Technical report. University of Pennsylvania [a.o.].
- Ellen Prince. 1999. How not to mark topics: ‘Topicalization’ in English and Yiddish. *8 Texas Linguistics Forum*.
- Tanya Reinhart. 1980. Conditions for text coherence. *Poetics Today* 1(4): 161–180.
- Tanya Reinhart. 1982. *Pragmatics and linguistics: An analysis of sentence topics*. Reprint of an earlier publication in 1981, Indiana University Linguistics Club.
- Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In D. H. Jones and H. Somers (eds.) *New Methods in Language Processing*, UCL Press, 154–164.
- Thomas Schmidt. 2004. EXMARaLDA – ein Modellierungs- und Visualisierungsverfahren für die computergestützte Transkription gesprochener Sprache. In *Proceedings of Konvens*. Vienna, Austria.
- Augustin Speyer. 2005. Competing Constraints on Vorfeldbesetzung in German. In *Proceedings of Constraints in Discourse Workshop*, 79–87. Dortmund, Germany.
- Augustin Speyer. 2007. Die Bedeutung der Centering Theory für Fragen der Vorfeldbesetzung im Deutschen. *Zeitschrift für Sprachwissenschaft*, 26: 83–115.

This book contains state-of-the-art contributions to the 10th conference on Natural Language Processing, KONVENS 2010 (Konferenz zur Verarbeitung natürlicher Sprache), with a focus on semantic processing.

The KONVENS in general aims at offering a broad perspective on current research and developments within the interdisciplinary field of natural language processing. The central theme draws specific attention towards addressing linguistic aspects of meaning, covering deep as well as shallow approaches to semantic processing. The contributions address both knowledge-based and data-driven methods for modelling and acquiring semantic information, and discuss the role of semantic information in applications of language technology.

The articles demonstrate the importance of semantic processing, and present novel and creative approaches to natural language processing in general. Some contributions put their focus on developing and improving NLP systems for tasks like Named Entity Recognition or Word Sense Disambiguation, or focus on semantic knowledge acquisition and exploitation with respect to collaboratively built resources, or harvesting semantic information in virtual games. Others are set within the context of real-world applications, such as Authoring Aids, Text Summarisation and Information Retrieval. The collection highlights the importance of semantic processing for different areas and applications in Natural Language Processing, and provides the reader with an overview of current research in this field.