Text Resources and Lexical Knowledge: Selected Papers from the 9th Conference on Natural Language Processing, KONVENS 2008

Edited by
Angelika Storrer et al.

Mouton de Gruyter

Text Resources and Lexical Knowledge

Text, Translation, Computational Processing

8

Editors
Annely Rothkegel
John Laffling

Mouton de Gruyter
Berlin · New York

Text Resources and Lexical Knowledge

Selected Papers from the 9th Conference on Natural Language Processing KONVENS 2008

Edited by
Angelika Storrer
Alexander Geyken
Alexander Siebert
Kay-Michael Würzner

Mouton de Gruyter
Berlin · New York

Mouton de Gruyter (formerly Mouton, The Hague)
is a Division of Walter de Gruyter GmbH & Co. KG, Berlin.

Library of Congress Cataloging-in-Publication Data

Konferenz „Verarbeitung Natürlicher Sprache“ (9th : 2008)
Text resources and lexical knowledge : selected papers from the 9th Conference on Natural Language Processing, KONVENS, 2008 / edited by Angelika Storrer ... [et al.].
p. cm. – (Text, translation, computational processing ; 8)
Includes bibliographical references and index.
ISBN 978-3-11-020735-4 (hardcover : alk. paper)
1. Lexicology – Data processing – Congresses. 2. Discourse analysis – Data processing – Congresses. 3. Corpora (Linguistics) – Congresses. I. Storrer, Angelika. II. Title.
P326.5.D38K66 2008
413.0285–dc22
2008029228

∞ Printed on acid-free paper which falls within the guidelines
of the ANSI to ensure permanence and durability.

ISBN 978-3-11-020735-4
ISSN 1861-4272

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Preface

The 9th biennial conference on Natural Language Processing (KONVENS 2008) will be held from September 30th to October 2nd 2008 at the "Berlin-Brandenburg Academy of Sciences and Humanities" (BBAW) in Berlin.

The central topic of this conference is the dynamic interaction between digital text resources and lexical knowledge representations. The conference papers collected in this volume describe innovative approaches to various aspects of this interaction in three different sections.

- The papers in the section "Linguistic Analysis of Text Resources" discuss techniques, tools and models for the automated linguistic analysis of various types of text resources (e.g. tree banks, historical and present-day text corpora).
- The papers in the section on "Extraction of Lexical Knowledge from Text Resources" describe and evaluate methods to automatically extract monolingual and bilingual lexical knowledge from text resources (e.g. statistical and rule-based methods, machine learning, "hybrid" approaches).
- The papers in the section "Representation of Lexical Knowledge" present innovative approaches to represent lexical knowledge in digital media for different purposes and user groups (e.g. natural language processing, information extraction, lexical information systems for human users, lexical resources for research and teaching)

The contributions in this volume - as well as the poster and demonstration papers published in a separate volume - provide a substantial overview of current trends and issues in the fields of computational lexicography and lexicology, corpus linguistics and text technology.

The Konferenz zur Verarbeitung natürlicher Sprache ("conference on natural language processing", KONVENS) was initiated in 1992 by the academic societies DEGA, DGfS, GI, GLDV, ITG and ÖGAI, each of which takes turns organizing the conference. The 9th annual Konvens 2008 is hosted by the GLDV (Gesellschaft für linguistische Datenverarbeitung) in collaboration with the "DWDS" project team (Digital Dictionary of the German Language) belonging to the Center of Language (Zentrum Sprache) at the Berlin-Brandenburg Academy of Sciences and Humanities (BBAW).

Angelika Storrer, Alexander Geyken, Alexander Siebert, Kay-Michael Würzner
September 2008

Acknowledgments

The conference has kindly been sponsored by the "Hermann und Elise geborene Heckmann Wentzel-Stiftung" and by the GLDV. We would like to thank both sponsors. Our gratitude especially goes out to Wolfgang Klein and Christiane Fellbaum, who have shown us valuable support during all phases of planning and putting on the conference.

We would also like to thank the members of the Programme Committee and the members of the Review Board for providing detailed and fair reviews in due time. Last but not least we would like to thank Marko Drotschmann, Amanda Hicks, Laura Hope and Bryan Jurish, who all helped format and edit this volume.

Programme committee

Ernst Buchberger (ÖGAI)
Stefan Busemann (GI)
Alexander Geyken (BBAW)
Jost Gippert (BBAW)
Karin Harbusch (GI)
Rüdiger Hoffmann (ITG)
Bernd Kröger (DEGA)
Lothar Lemnitzer (GLDV)
Sabine Schulte im Walde (DGFS-CL)
Manfred Stede (GLDV)
Angelika Storrer (GLDV, BBAW)

Review board

Melina Alexa (BIFAB Mannheim)
Irene Cramer (Dortmund)
Stefan Engelberg (IDS Mannheim)
Christiane Fellbaum (Princeton, BBAW)
Anette Frank (Heidelberg)
Iryna Gurevych (Darmstadt)
Thomas Hanneforth (Postdam)
Uli Heid (Stuttgart)
Anke Holler (Göttingen)
Wolfgang Höppner (Duisburg-Essen)
Bryan Jurish (BBAW)
Henning Lobin (Gießen)
Harald Lüngen (Gießen)
Alexander Mehler (Bielefeld)
Gerald Neumann (BBAW)
Manfred Pinkal (Saarbrücken)
Thomas Schmidt (Hamburg)
Roman Schneider (IDS Mannheim)
Bernhard Schröder (Uni Duisburg-Essen)
Elke Teich (Darmstadt)
Andreas Witt (Tübingen)
Heike Zinsmeister (Heidelberg)

Contents

Preface v

I Linguistic analyses of text resources

Extending the TIGER query language with universal quantification 3
Torsten Marek, Joakim Lundborg and Martin Volk

Exploring automatic theme identification: a rule-based approach 15
Lara Schwarz, Sabine Bartsch, Richard Eckart and Elke Teich

Finding canonical forms for historical German text 27
Bryan Jurish

Computing distance and relatedness of medieval text variants from German 39
Stefanie Dipper and Bettina Schrader

Data structures for the analysis of regional language variation 53
Birgit Kellner, Timm Lehmberg, Ingrid Schröder and Kai Wörner

Local syntactic tagging of large corpora using weighted finite state transducers 65
Jörg Didakowski

II Extraction of lexical knowledge from text resources

Towards improved text understanding with WordNet 81
Christiane Fellbaum, Peter Clark and Jerry Hobbs

Influence of accurate compound noun splitting on bilingual vocabulary extraction 91
Marcin Junczys-Dowmunt

A taxonomy of task-related dialogue actions: the cases of tutorial and collaborative planning dialogue 105
Magdalena Wolska and Mark Buckley

Visualization of dialect data 119
Erhard Hinrichs and Thomas Zastrow

Providing corpus data for a dictionary for German juridical phraseology 131
Ulrich Heid, Fabienne Fritzinger, Susanne Hauptmann, Julia Weidenkaff and Marion Weller

A tool for corpus analysis using partial disambiguation and bootstrapping of the lexicon 145
Kurt Eberle, Ulrich Heid, Manuel Kountz and Kerstin Eckart

Rapid construction of explicative dictionaries using hybrid machine translation 159
Kurt Eberle and Reinhard Rapp

III Representation of lexical knowledge and text resources

The spanish version of WordNet 3.0 175
Ana Fernández-Montraveta, Gloria Vázquez and Christiane Fellbaum

An OLIF-based open inflectional resource and yet another morphological system for German 183
Simon Clematide

Tools for exploring GermaNet in the context of cl-teaching 195
Irene Cramer and Marc Finthammer

Adaptive word sense views for the dictionary database eWDG: The case of definition assignment 209
Axel Herold and Alexander Geyken

Research on dictionary use and the development of user-adapted views 223
Carolin Müller-Spitzer

The Kicktionary revisited 239
Thomas Schmidt

List of authors 252

Index 259

Part I
Linguistic analyses of text resources

Extending the TIGER query language with universal quantification

Torsten Marek, Joakim Lundborg and Martin Volk

Abstract. The query language in TIGERSearch is limited due to its lack of universal quantification. This restriction makes it impossible to make simple queries like “Find sentences that do not include a certain word”. We propose an easy way to formulate such queries. We have implemented this extension to the query language in a tool that allows for querying parallel treebanks, while including their alignment constraints. Our implementation of universal quantification relies on the view of node sets rather than single node unification. Our query tool is freely available.

1 Introduction

In the last ten years, many languages and tools for querying syntactically annotated corpora have been developed. Most of these tools and query languages have been designed for a specific corpus and a specific annotation format. TGrep2[1] was originally designed for the Penn Treebank and its tree format. TIGERSearch (König and Lezius 2003; Lezius 2002) was designed for the NEGRA and TIGER corpora (Brants et al. 2002) and uses the TIGER-XML format.

TIGERSearch is a tool that allows the user to search a treebank by formulating queries in feature-value pairs. For example, one may search for the word ‘can’ with the part-of-speech ‘noun’ by querying

[word="can" & pos="NN"] (1.1)

In addition to constraints over words, the TIGER query language allows the user to also use constraints over dominance relations (search for a node *n1* that dominates a node *n2* in the tree), precedence relations and node predicates (such as arity, discontinuous and root).

In general, the design of the input format influences to a large extent the design of the query language, since it defines what can be queried. For instance, TIGER-XML supports crossing branches, leading to non-terminal nodes whose terminals are not a proper substring of the sentence (discontinuous nodes). The TIGER query

[1] `http://tedlab.mit.edu/~dr/TGrep2/`

language thus has special functions for dealing with discontinuous nodes. In contrast, the Penn Treebank formalism does not support crossing branches directly, and thus TGrep2 has no means to handle this notion.[2]

While certain limitations of query languages are thus due to the original design and could only be approximated, other interesting queries may simply be missing from the query language. Lai and Bird (2004) list seven example queries, named Q1-Q7, which each formalism should support, regardless of the annotation formalism[3].

In this paper, we will deal with queries that require universal quantification, i.e. selecting a tree by stating constraints over possibly many nodes rather than individual nodes. The sample queries contain two examples where universal quantification is needed (Lai and Bird 2004: p. 2):

Q2. Find sentences that do not include the word ‘saw’.

Q5. Find the first common ancestor of sequences of a noun phrase followed by a verb phrase.

With the TIGER query language and its implementation TIGERSearch, these queries can only be approximated. The result set generated for the approximated queries will likely contain errors.

In section 2, we will analyze the limitations of TIGER and TIGERSearch and look at the solution proposed in Lezius (2002). In section 3, we will develop an extension to the TIGER query formalism that is able to deal with Q2 and similar queries, and we will analyze the different requirements of Q2 and Q5 in section 5.

In section 4, we will talk about the implementation of the extensions from the previous section in our own implementation of the TIGER query language, which is part of the Stockholm TreeAligner[4] (Volk et al. 2007).

In the remainder of the paper, we will speak of syntax *graphs* rather than trees. These graphs are acyclic, directed and do not contain structure sharing (i.e. each node has exactly one direct ancestor). However, due to the existence of crossing branches, TIGER trees cannot be stored as nested lists or XML DOM trees directly, which is the usual format of trees.

Node descriptions are boolean expressions of feature constraints of the form "[feature=value]". They are used for finding nodes (assignments) in the corpus which are then used for the constraint resolution in TIGER queries.

[2] Crossing branches should not be confused with secondary edges. While crossing branches lead to crossing edges in the primary level of a graph, secondary edges constitute a secondary level.

[3] with one exception: Q6 assumes multiple layers of annotation.

[4] `http://dev.ling.su.se/treealigner`

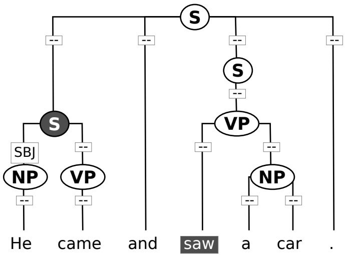


Figure 1. Wrong highlights.

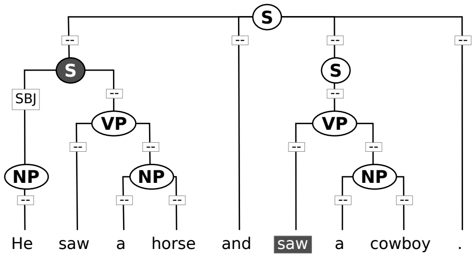


Figure 2. A false positive.

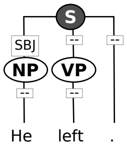


Figure 3. A false negative.

2 Limitations of the TIGER query language

Figures 1 to 3 show three example graphs for Q2 ("Find sentences that do not include the word 'saw' "). If the query were evaluated correctly, the result set would only contain the first coordinated sentence from figure 1 ("He came") and the main sentence of figure 3 ("He left").

In the TIGER query language, however, every node variable is implicitly existentially quantified. For example, the query

#s:[cat="S"] !>* #w:[word="saw"] (1.2)

states that some node of category S does not dominate some word with the surface string 'saw'. This query is interpreted by TIGERSearch so that it returns all combinations of two nodes #s, #w[5] in all graphs, such that #s does not dominate #w.

[5] the hash symbol '#' is used to mark variables

From the graphs that were actually meant to be matched by Q2, it will only return those that have a terminal ‘saw’ outside of any S node. Thus, the result set contains the graph from figure 1 (with extra highlighting, which is distracting, but not fatal), but also, for instance, the graph from figure 2, which should definitely not be matched by the query. But all graphs that do not contain ‘saw’, like the one in figure 3, will not show up in the result set.

Another attempt to formulate Q2 in TIGERSearch is the query

#s:[cat="S"] >* #w:[word!="saw]" (1.3)

which states that some node of category S dominates some terminal that does not have the surface string ‘saw’. Looking at the example graphs, it immediately becomes clear that there are many combinations of nodes that satisfy this query.

In general, there is no correct way to formulate the desired query in TIGERSearch. This limitation is acknowledged by the developers:

> The use of the universal quantifier causes computational overhead since universal quantification usually means that a possibly large number of copies of logical expressions have to be produced. For the sake of computational simplicity and tractability, the universal quantifier is (currently) not part of the TIGER language. (TIGERSearch Help, section 10.3)

Section 5.7 of Lezius (2002) contains a proposal for an extension of the TIGER query formalism that combines universal quantification and the implication operator. Using this syntax, Q2 can now be formulated as

$$\forall\text{#w}\,(\text{[cat="S"]} >^{*} \text{#w}) \Rightarrow \text{#w:[word!="saw"]} \tag{1.4}$$

which means that every node dominated by an S node must not be the surface string ‘saw’.

Expressing the queries with the implication operator would be natural for the unification-based evaluation of queries in TIGERSearch. As already mentioned, an actual implementation comes at great computational cost. For each ∀ clause in the query, all nodes in the graph have to be iterated to find out if they satisfy $l \Rightarrow r$. In some cases, using the logical equivalent $!r \vee l$ can be used to speed up the queries. For instance, Q2 can simply be evaluated by

1. retrieving all S nodes in all graphs.

2. retrieving all nodes where [word="saw"].

3. for each S node, checking if it dominates none of the nodes from step 2.

Apart from runtime complexity considerations, the syntax would be extended with a construct that is conceptually hard to grasp and that makes the grammar of the query language much more complex. We therefore decided to explore a different path.

3 Design of the universal quantification extension

The Lezius solution presented in section 2 builds on the query calculus that is at the core of TIGERSearch's query evaluation engine. This calculus is based on unifying the partial graph description given by the query with any of the graph definitions in the corpus. If the unification succeeds, the graph matches the query and is returned in the result set.

In contrast, the query engine in the Stockholm TreeAligner is based on node sets, and combinations of nodes from the different sets to satisfy the constraints given in a query[6].

In the previous analysis of Q2, we showed that it is possible to rephrase the query using logical equivalents. Therefore, the query "get all S nodes that do not contain the word saw" can be rephrased into "get all graphs where all instances of 'saw', if any, are not dominated by a specific S node".

This is essentially expressed in query 1.2. But as we already showed, the usage of the existential quantification will not lead to the expected results. However, if one of the two operands is not understood to be a single node from the graph, but a *set* of nodes, the result will be correct. Therefore, we introduce a new data type into the query language, the node set which we indicate with the % symbol. Bare node descriptions are still bound by an implicit existential quantifier as before. A node set is only bound to a variable that starts with a percentage symbol:

```
#s:[cat="S"] !>* %w:[word="saw"]          (1.5)
```

If one operand in a constraint is a node set instead of a node, the semantics of the constraint are changed. Here, only those assignments to #s are returned where the constraint '!>*' holds for each node in the node set %w, which contains all terminals with the surface string 'saw'. When applied to the graphs from the small sample corpus, this query now does not yield any false positives like the graph in figure 2.

The semantics of node predicates as defined in the TIGER query language do not change; they still operate at the node level. In the query

```
%np:[cat="NP"] & tokenarity(%np, 2)          (1.6)
```

[6] cf. section 4 for a brief outline of the evaluation strategy.

the node set %np will contain all NPs whose token arity is 2 (which means that each NP dominates exactly two tokens). See section 5 for a further discussion.

3.1 Node sets

If each variable is bound by an existential quantifier, evaluation of a query can terminate as soon as one node description does not yield any results. Graphs that do not contain matching nodes for any of the descriptions will also be disregarded, which is why the graph from figure 3 will still not be matched by the query. To produce correct results, the semantics of node descriptions bound to node sets have to be changed. In contrast to existentially quantified nodes, which may not be undefined, a set can be the empty set $\emptyset$. If this is the case, a constraint is trivially true.

With this change in place, TIGER is in Cantor's paradise, and no one shall expel it from there. The basic semantics of set types are defined and new set predicates can be introduced to make set queries more powerful. As an example, consider the query "Return all NPs that do not contain any PP, but only if the graph contains PPs". Given that empty node sets are now allowed, the query has to be written as

[cat="NP"] !>* %pp:[cat="PP"] & [cat="PP"]. (1.7)

The last term ensures that at least one PP exists in the graph. As a side effect, the result set will contain one entry for each combination of NP and PP in a matching graph, which is slightly more than what the query was supposed to yield.

To express constraints on set cardinality, a syntax for set algebra operations could be added to TIGER. As an example, to make sure that the set %pp contains elements, one could think of something like

[cat="NP"] !>* %pp:[cat="PP"] & {size(%pp) > 0} (1.8)

Any expression enclosed in curly brackets is evaluated as an operation on sets. This addition would add a lot of power to the query language, but would make it much harder to use. Also, it requires a nontrivial amount of implementation effort and makes the query grammar more cumbersome. In our opinion, node set operations go beyond the scope of what can conveniently be handled inside of a single expression. If these expressions were added, it would be desirable to store node sets, reuse them in later queries or combine node sets from different queries.

Instead of adding full support for set operations, we introduce two new predicates that operate exclusively on node sets: *empty* and *nonempty*. The semantics of the predicates can be inferred from the names, and the previous query can be written in a straightforward manner:

[cat="NP"] !>* %pp:[cat="PP"] & nonempty(%pp) (1.9)

It is now also possible to search for graphs that do not contain a specific kind of node by using *empty*. The query

%w:[pos="det"] & empty(%w) (1.10)

returns all graphs that do not contain any determiner.

4 Implementation of the query language within the Stockholm TreeAligner

We have re-implemented the TIGER query language in a tool for the creation and exploration of parallel treebanks, the Stockholm TreeAligner. The TreeAligner allows the user to load two treebanks, typically with parallel (i.e. translated) sentences. For example, we have used the TreeAligner to work on a German treebank and its parallel English treebank. We have aligned the two treebanks first on the sentence level to get corresponding tree pairs and then on the word level and phrase level. Figure 4 shows a tree pair from our parallel treebank.

We currently distinguish between exact translation correspondence, represented by green lines, and approximate ('fuzzy') translation correspondence represented by red lines. Although it is useful in principle to make this distinction, applying it consistently in practice has proven to be difficult.

The TreeAligner is an editor which allows the user to create and modify alignments. It can also be used to browse and search parallel treebanks, which is where the query language comes in. We have re-implemented and extended the TIGER query language within the TreeAligner to search both treebanks in parallel. We have also added alignment constraints in order to combine the queries. Let us consider the following example query:

```
treebank1    #node1:[cat="NP"] > [cat="PP"]
treebank2    #node2:[cat="NP"] > [cat="PP"]        (1.11)
alignment    #node1 * #node2
```

Here, #node1 is a variable that identifies a node of category NP in a graph in treebank1. Likewise, #node2 identifies a node of category NP in *treebank2*. These variables correspond exactly to the syntax in the TIGERSearch query language. We then use these variables in the alignment query. The general alignment relation is indicated by the '*' operator. For a detailed description of the alignment query syntax, see Volk et al. (2007). This query searches through both *treebank1* and *treebank2* for noun phrases that dominate a prepositional phrase and returns all matches where these noun phrases are aligned with each other.

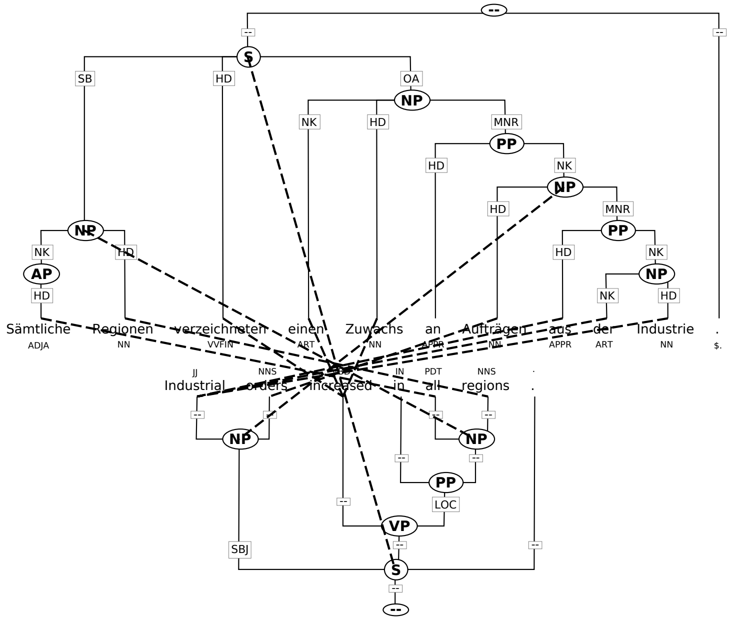


Figure 4. An example of two aligned sentences from the SMULTRON corpus

The implementation of the TIGER query language in the Stockholm TreeAligner is based on sets of nodes and constrained Cartesian products over these sets. Because of that, it was possible to implement the extensions described in section 3 with little effort. In contrast to existentially quantified nodes, node sets are subject to some restrictions. In a constraint, at most one operand may be a node set. Constraints that have two node sets as operands will lead to a runtime error[7]. In the result display, in contrast to existentially quantified nodes, nodes from sets are not highlighted. While there are no technical reasons for this, having node sets stand out from the rest is not helpful, simply because too many nodes would be highlighted. This would lead to a confusing result display.

[7] This behavior may change in the future. However, to this date we do not have linguistically interesting queries that require a constraint with two node sets.

5 Beyond node descriptions

Using the implication operator, Q5 (finding the first common ancestor of a sequence NP VP) can be expressed in the following manner (where NT stands for any non-terminal):

```
#a:[NT] >* #np:[cat="NP"] &
#a >* #vp:[cat="VP"] &
#np .* #vp &                                   (1.12)
#np !>* #vp & #vp !>* #np &
∀#b (#a >* #b & #b >* #np) ⇒ (#b !>* #vp)
```

The query looks for a node #a that dominates a sequence of nodes #np #vp, but only if no node on the path from #a to #np (the condition of the ∀ clause) also dominates #vp. The fourth line of the query is needed because one of the nodes #np, #np could dominate the other one, as well as precede it, given that its left corner precedes the left corner of the other one. With these restrictions in place, it is made sure that #a really is the first common ancestor.

It is clear that this query cannot be handled by the node set definition syntax, because, unlike in Q2, the right-hand side of the implication tests for a non-local feature of the node.

For these more complex queries, we propose an extension that is functionally equivalent to the implication syntax, but also relies on the notion of node sets. If the right-hand side of a node set definition is enclosed in curly brackets, the contents of the definition are interpreted as a *sub-query* rather than a node description, which are enclosed in square brackets. In this sub-query, all existential node variables from the outer query may be referenced. In the example query, the node set of all nodes on the path between #a and #np would be defined as

```
%b:{#a >* %:[NT] & % >* #np}.                  (1.13)
```

The free-standing percent sign is used as the placeholder for those nodes that are going to be elements of the set[8]. With the node set %b defined this way, query 1.12 can be rephrased as

```
#np:[cat="NP"] .* #vp:[cat="VP"] &
#a:[NT] >* #np &
#a >* #vp &                                    (1.14)
#np !>* #vp & #vp !>* #np &
%b:{#a >* %:[NT] & % >* #np} !>* #vp
```

[8] This syntax is still subject to change.

While this query will need more time for evaluation, the evaluation order is essentially unchanged:

1. Get all nodes that can match #np, #vp, #a and %.

2. For a given assignment, execute the sub-query that defines %b.

3. Evaluate all other constraints.

While this approach is easy to implement, it is also very slow. Greater speeds can be achieved by early pruning of the search space, for instance by employing additional indices or intelligent constraint evaluation reordering and caching.

With sub-queries, the semantics of node predicates should be changed as well. A node predicate that is invoked on a set of nodes will only return true if all nodes in the set satisfy the predicate. To build the set of all nodes that fulfill a certain condition, a sub-query should be used. For instance, the set of all NP nodes with exactly two children can be created with

$$\%\text{x:}\{\%\text{:[cat="NP"] \& arity(}\%\text{, 2)}\} \tag{1.15}$$

6 Related research

The Stockholm TreeAligner is unique in its ability to display parallel syntax trees and their phrase alignment. It can be seen as an advancement over tools like Cairo (Smith and Jahr 2000) or I*Link (Merkel et al. 2003), which were developed for creating and visualizing word alignments but are unable to display trees. The TreeAligner is also related to Yawat and Kwipc (Germann 2007), two software tools for the creation and display of sub-sentential alignments in matrices.

Our implementation of the query language owes a lot to TIGERSearch, whose ground-breaking work and robust implementation we gratefully acknowledge. But our work is also related to Emdros (Petersen 2004), which has its own system for parallel corpus searches. In Petersen (2005), the author of Emdros shows that his system can handle Q2 correctly and is faster than TIGERSearch. Other related query systems are NXT search/NiteQL (Evert and Voormann 2003; Heid et al. 2004) and LPath (Bird et al. 2005), none of which is specifically geared towards parallel treebanks.

Extending the TIGER query language with set logic operations has been proposed and implemented in Saers (2006), however in a different manner. The results of several queries can be combined with set operations, which allows one to express queries which need a universal quantifier if expressed in a single query.

LPath is an extension of XPath to support querying syntactic trees that are encoded in XML info sets. All syntactical relations (dominance, precedence) are directly encoded in the XML structure and thus inherit the restrictions that apply to XML trees. Secondary layers of annotation, even simple ones like the secondary edge feature in TIGER-XML, are not supported. LPath also provides an extension to the XPath syntax allowing `not` modifiers on predicates, making it possible to express Q2 in a straight-forward manner.

NiteQL has been designed for corpora with multiple layers of annotation on one and the same text, but also heavily relies on XML to encode syntactical structure, with the same restrictions. According to Lai and Bird (2004), NiteQL cannot handle queries that need universal quantification. The authors of NiteQL also say

> TigerSearch (sic) has a nice graphical interface and again supports structural operators missing in NQL. (Nite XML Toolkit Documentation, section 8.4)

but do not give any examples for the differences.

7 Conclusion and outlook

We have shown how the TIGERSearch query language has been integrated in a parallel treebank exploration tool, the Stockholm TreeAligner. This tool allows for parallel querying over two treebanks combined with their alignment constraints and is thus a useful tool for cross-language comparisons and translation studies (cf. Lundborg et al. (2007) for an introduction).

The TIGERSearch query language lacks universal quantification and is thus unable to handle certain queries. We have added this functionality to the query language and implemented it in the TreeAligner. In this paper, we have also sketched a possible syntax for queries that contain more complex universal quantification.

The extensions described in section 3 will be part of the next release of the Stockholm TreeAligner later this year, and are already implemented in the development version. This release will also contain a browser and query interface for monolingual treebanks, similar to the one provided by TIGERSearch. While the functionality for monolingual queries is already (and has always been) provided by the underlying implementation, so far the focus of the TreeAligner has been on creating and querying parallel treebanks.

We will also follow up on the planned extensions sketched in section 5. However, the TIGER implementation in the Stockholm TreeAligner is still in a very early stage. Feature completeness and a faster query evaluation will become more important for the upcoming releases. We will also provide data on the efficiency of queries with universal quantification as well as a comparison of query evaluation speed in both the Stockholm TreeAligner and TIGERSearch.

8 Acknowledgements

We would like to thank Armin Schmidt, Lisa Hunemark and our three anonymous reviewers for helpful comments and corrections.

References

Bird, Steven, Yi Chen, Susan Davidson, Haejoong Lee, and Yifeng Zheng (2005). Extending XPath to Support Linguistic Queries. In *Proc. of Programming Language Technologies for XML (PLANX)*, 35–46, Long Beach, California.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, URL `citeseer.ist.psu.edu/brants02tiger.html`.

Evert, Stefan and Holger Voormann (2003). Nql – a query language for multi-modal language data. Technical report, IMS, University of Stuttgart, Stuttgart.

Germann, Ulrich (2007). Two Tools for Creating and Visualizing Sub-Sentential Alignments of Parallel Text. In *Proc. of The Linguistic Annotation Workshop at ACL 2007*, 121–124, Prague.

Heid, Ulrich, Holger Voormann, Jan-Torsten Milde, Ulrike Gut, Katrin Erk, and Sebastian Padó (2004). Querying Both Time-Aligned and Hierarchical Corpora with NXT Search. In *Proc. of The Fourth Language Resources and Evaluation Conference*, Lisbon.

König, Esther and Wolfgang Lezius (2003). The TIGER language – A Description Language for Syntax Graphs, Formal Definition. Technical report, IMS, University of Stuttgart.

Lai, Catherine and Steven Bird (2004). Querying and Updating Treebanks: A Critical Survey and Requirements Analysis. In *Proceedings of the Australasian Language Technology Workshop*.

Lezius, Wolfgang (2002). *Ein Suchwerkzeug für Syntaktisch Annotierte Korpora*. Ph.D. thesis, IMS, University of Stuttgart.

Lundborg, Joakim, Torsten Marek, Maël Mettler, and Martin Volk (2007). Using the Stockholm TreeAligner. In *Proc. of The 6th Workshop on Treebanks and Linguistic Theories*, Bergen.

Merkel, Magnus, Michael Petterstedt, and Lars Ahrenberg (2003). Interactive Word Alignment for Corpus Linguistics. In *Proc. of Corpus Linguistics 2003*, Lancaster.

Petersen, Ulrik (2004). Emdros: A Text Database Engine for Analyzed or Annotated Text. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, 1190, Morristown, NJ, USA: Association for Computational Linguistics, doi:http://dx.doi.org/10.3115/1220355.1220527.

Petersen, Ulrik (2005). Evaluating Corpus Query Systems on Functionality and Speed: TIGERSearch and Emdros. In *Proc. of The International Conference on Recent Advances in NLP*, Borovets.

Saers, Markus (2006). TiSS: Tiger set search.

Smith, Noah A. and Michael E. Jahr (2000). Cairo: An Alignment Visualization Tool. In *Proc. of LREC-2000*, Athens.

Volk, Martin, Joakim Lundborg, and Maël Mettler (2007). A Search Tool for Parallel Treebanks. In *Proc. of The Linguistic Annotation Workshop (LAW) at ACL*, Prague.

Exploring automatic theme identification: a rule-based approach

Lara Schwarz, Sabine Bartsch, Richard Eckart and Elke Teich

Abstract. Knowledge about Theme-Rheme serves the interpretation of a text in terms of its thematic progression and provides a window into the topicality of a text as well as text type (genre). This is potentially relevant for NLP tasks such as information extraction and text classification. To explore this potential, large corpora annotated for Theme-Rheme organization are needed. We report on a rule-based system for the automatic identification of Theme to be employed for corpus annotation. The rules are manually derived from a set of sentences parsed syntactically with the Stanford parser and analyzed in terms of Theme on the basis of Systemic Functional Grammar (SFG). We describe the development of the rule set and the automatic procedure of Theme identification and assess the validity of the approach by application to some authentic text data.

1 Introduction

Text data applications of NLP, such as information extraction (IE) or document classification (DC), require a new look at issues of discourse parsing. While the focus in discourse parsing has been on qualitative analyses of *single* texts – for instance identifying the meaningful, coherent parts of a text (generic structure, rhetorical structure, logical structure; see e.g., Marcu (2000); Poesio et al. (2004)), interpreting reference relations (co-reference resolution) or analyzing information structure (e.g. Postolache et al. 2005) – the attention of NLP in IE/DC is on *sets* of texts and quantitative features. So far, the potential contribution of discourse knowledge for IE/DC applications has hardly been explored, since the predominant methods are string or word-based and even supervised data mining rarely employs information at higher levels of linguistic abstraction. Here, the bottleneck is often the lack of (large enough) corpora annotated in terms of discourse features.

The work reported on in this paper is placed in the context of enhancing corpora with linguistic features of discourse organization, an increasingly active research area (see e.g. Lobin et al. 2007; Lüngen et al. 2006; Stede and Heintze 2004). We report on the derivation of rules for automatic Theme identification from a set of sample sentences instantiating the principal Theme options of English. Our approach combines automatic syntactic parsing with annotation of Theme (cf. the work by Honnibal and Curran (2007) on enhancing the Penn Treebank in terms of features from Systemic Functional Grammar Halliday (2004), or Buráňová et al. (2000) on annotating corpora in terms of Topic-Focus articulation). Even if automatic Theme identification may not achieve 100 % accuracy (and manual postediting might be needed), corpora annotated for Theme would clearly be very useful as input for various machine learning applications as well as for linguists wishing to explore patterns of thematic progression in texts.

The paper is organized as follows. In Section 2 we present the underlying definition of Theme-Rheme employed in our work. Section 3 explains the experimental setting and presents the types of rules derived from the syntactic structure representation as well as the procedure for rule application. Section 4 discusses some results. We conclude with a summary and issues for future work (Section 5).

2 Definition of theme in english

According to Systemic Functional Grammar (Halliday 1985: 38), the Theme is defined as the point of departure of the clause as message. It is the contextual anchor of the clause that is oriented towards the preceding discourse (Matthiessen 1995: 531). The rest of the clause is the Rheme. In English, the Theme occupies the first constituent position in the clause:

> "As a general guide to start off with, we shall say that the Theme of a clause is the first group or phrase that has some function in the experiential structure of the clause." (Halliday 2004: 66)

Themes can be either simple (consisting of one Thematic element) or complex (consisting of multiple Thematic elements). Figure 1 displays the major simple Theme options for English declarative clauses. For some examples illustrating the most common options see examples 1-5 below (Themes are underlined).

1. *The duke gave my aunt this teapot.* [unmarked; nonpredicated, nonsubstitute]
2. *On Saturday night, I lost my wife.* [marked, circumstantial; nonpredicated, nonsubstitute]
3. *What the duke gave to my aunt was a teapot.* [unmarked; nonpredicated, substitute]
4. *It was a teapot that the duke gave to my aunt.* [unmarked; predicated]
5. *It was on Saturday that I lost my wife.* [marked, circumstantial; predicated]

For clause moods other than declarative (i.e. interrogative, imperative), the options are partly overlapping and partly mood-specific (see examples 6-7).

6. *Give him that teapot!* [imperative]
7. *Could you give him that teapot?* [interrogative]

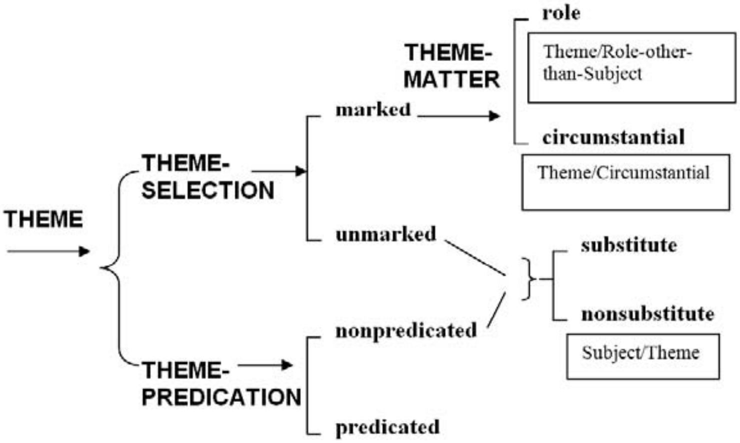


Figure 1. System network for Theme

Apart from these basic options with only one constituent filling the Theme position, there are also multiple Themes containing more than one constituent. For some examples, see 8-9 below.

8. *Aesthetically, in terms of the vision in your head, what is the relationship between the fiction and the non-fiction?* (textual Themes)
9. *Well Jane think of smoked salmon.* (textual and interpersonal Themes)

In these examples, we have topical Themes, *what; think*, the constituents of which at the same time have textual and interpersonal function respectively: *Aesthetically; Well; Jane*.

3 Patterns, rules and rule interpretation

The SFG definition of Theme is based entirely on clause syntax, which means that the syntax can be used as the only criterion to identify the Theme. The basis for the identification of syntactic patterns matching Theme is provided by the Stanford PCFG Parser (Klein and Manning 2002), a probabilistic parser with flexible output (parts of speech, phrase structure, dependency structure). For parsing results for examples 1 and 2 see Figures 2 and 3. Below, we describe the syntactic patterns found matching the different Theme types (Section 3.1) and present the procedure implemented for Theme identification (Section 3.2).

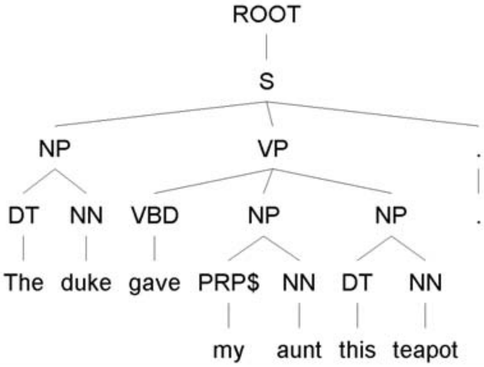


Figure 2. The duke gave my aint this teapot.

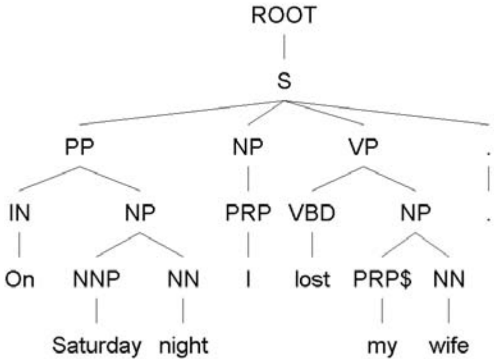


Figure 3. On Saturday night, I lost my wife.

3.1 Syntactic patterns and theme types

Manual Theme identification comprises two steps:

- determining the boundaries of the Theme in terms of syntactic phrases;
- assigning a Theme type to the unit identified.

In the simplest case, the Theme is the left-most constituent immediately dominated by the S node in the parse tree (see Figures 2 and 3). This is, in fact, also the most frequent case in English (Subject-Theme, Circumstance-Theme; see again examples 1 and 2 above). However, there are many syntactically more complex Themes, as shown in examples 8-9 above. Also, in order to assign a Theme type, we must take into consideration the mood of the clause (declarative, imperative or interrogative, see examples 6-7). For each mood type, several different syntactic struc-

Table 1. Simple Theme patterns and examples

#	Theme Type	Example	Pattern
1	Unmarked Subject	*The duke has given my aunt that teapot.*	(S(NP(*))(*)(VP(*))(*))
2	Existential There	*There were three jovial Welshmen.*	(S(NP(EX))(VP(*))(*))
3	Unmarked Nominalization	*What the duke gave to my aunt was that teapot.*	(S(SBAR(*))(*)(VP(*))(*))
4	Marked Adjunct: Adverbial Phrase	*Merrily we roll along.*	(S(ADVP(*))(NP(*))(VP(*)))
5	Marked Adjunct: PP	*From house to house I went my way.*	(S(PP(*))(*)(VP(*)))
6	Marked: Object as Theme	*The teapot the duke has given my aunt.*	(S(NP(*))(NP(*))(VP(*)))
7	Marked Nominalization	*What they could not eat that night the Queen next morning fried.*	(S(SBAR(*))(NP(*))(VP(*)))
8	Exclamative	*How dreadful she sounds.*	(S(ADJP(*))(NP(*))(VP(*)))
9	Unmarked: WH-Question	*Who wants a glass of wine?*	(SBARQ(*))
10	Unmarked: Yes/No Interrogative	*Did you sleep okay?*	(SQ(*))
11	Marked: Inverted Clause	*On the right is it?*	(SINV(*))
12	Unmarked Imperative	*Turn it down.*	(S(VP(*)(*))
13	Unmarked Thematic Equative	*The one who gave my aunt that teapot was the duke.*	(S(NP(NP(*)(SBAR(*)))(VP(*))(*)))
14	Marked Thematic Equative	*That is the one I like.*	(S(NP(DT))(VP(SBAR(*)))(*))

tures are possible and for each of the possible structures, we can have an unmarked or marked Theme variety. Each combination of syntactic structure, mood and markedness/unmarkedness of Theme creates its own pattern (or set of patterns) to which a Theme type can be assigned. Identification of the Theme then amounts to specifying the possible patterns in terms of partial syntactic trees that map onto the different Theme types. On the basis of these patterns, a set of rules can be specified that match the partial trees (see Section 3.2). For a start, we focus on simple Themes.

The patterns and corresponding rules were developed using a set of 85 sentences taken from Halliday (2004: 65–81) which are considered representative of the different Theme types in English, but exhibit some syntactic variation. The examples were run through the parser and annotated manually for Theme according to Halliday's model. Clause mood, one of the factors in Theme identification, can be read off the parse tree so that the corresponding mood-specific Theme types can be accounted for. For example, we observe that a declarative sentence has a top node `S`, whereas an interrogative has either `SBARQ` or `SQ` as top node (see patterns in Table 1). Additionally, there are more general patterns that hold for all mood types. This includes, for instance, the options of marked Theme. This can be seen in example 10, the interrogative version of example 2 above, with an identical Theme (conflated with the Adjunct):

10. *On Saturday night, did you lose your wife?*

Altogether, we derived 14 patterns for simple Themes from the examples, which correspond to 14 rules (see Section 3.2). These are displayed in Table 1 together with examples.

3.2 Rules and the tree-rule processor

We map the parsing output to an XML representation as commonly used in work on corpora (Eckart 2006). We are thus building on modified output from the Stanford parser for our analysis. Thus, the formal basis for rule specification is an XML representation.

The tree-rule processor is a simple pattern matcher implemented in Java which applies rules defined in a rule file to a parse read from a parse file. Figure 4 shows a trimmed down version of a parse file. A parse file can contain any number of sentences. Even though the parser generates more information, we currently only use the category (`cat`) attribute in the rules.

A rule consists of a label and a pattern. The label denotes the Theme type to be assigned and is encoded in the `label` attribute of the `rule` element. The pattern is expressed as an XML fragment specifying those elements and attributes which must be present in the parse for the rule to match. It is encoded in the children of the `rule` element. Attributes not specified in the rule, but present in the parse, are ignored. If a particular branch of the parse is of no importance, this branch is expressed as a wildcard (`relax`) in the rule. Figure 5 shows the *Marked Adjunct: PP* rule that matches the parse in Figure 4. Generally, the rules apply in the order of more specific to more general. This is done by matching the rules against each other. If a rule matches another, the rule that matched is less specific.

```
<Constituent cat="S">
  <Constituent cat="PP">
    <Constituent cat="IN">On</Constituent>
    <Constituent cat="NP">
      <Constituent cat="NNP">Saturday</Constituent>
      <Constituent cat="NN">night</Constituent>
    </Constituent>
  </Constituent>
  <Constituent cat="NP">
    <Constituent cat="PRP">I</Constituent>
  </Constituent>
  <Constituent cat="VP">
    <Constituent cat="VBD">lost</Constituent>
    <Constituent cat="NP">
      <Constituent cat="PRP\$">my</Constituent>
      <Constituent cat="NN">wife</Constituent>
    </Constituent>
  </Constituent>
  <Constituent cat=".">.</Constituent>
</Constituent>
```

Figure 4. Example XML parse output

4 Application and results

To test our approach, we carried out three experiments applying the rules to (a) the set of sample sentences used as base data to derive the patterns, (b) to a small set of texts and (c) to a larger corpus of 209 abstracts.

```
<rule id="rule 5" label="Marked Adjunct:PP">
  <Constituent cat="S">
    <!-- mark-theme -->
    <Constituent cat="PP"><relax/></Constituent>
    <!-- mark-theme -->
    <relax/>
    <Constituent cat="VP"><relax/></Constituent>
    <relax/>
  </Constituent>
</rule>
```

Figure 5. Rule: *Marked Adjunct: PP*

In order to measure the quality of the rules, the tree-rule processor may be run in a *training* mode. In the *training* mode, the input parse file may be annotated at any opening tag with an `expect` attribute naming the rule which is expected to match at this point. At most, one rule can be expected. Whenever a rule matches at an element that does not yet expect a rule, the user is prompted with a selection of all rules that matched. The user may then choose one of the rules as the "correct" one or choose none at all. The choices are saved back to the parse file. Using these "annotated" parse files, we can generate some statistics on the performance of the rules. The rest of this section will discuss the performance of the rules on the three sets of test data.

4.1 Set of sample sentences

Table 3 shows the performance statistics of the rules on the first data set. Here (and in subsequent tables), *sentences found* is the total number of full sentences provided to the tree-rule processor by the parser, *classified sentences* is the percentage of the total number of sentences for which a match was found and *total matches* represents the number of times a rule matched a sentence, clause or sentence fragment. *Matches MET*, or *true positives*, is the number of times an expected rule was matched and *matches UNEXPECTED* are instances when no rule was expected, yet a rule matched. This also includes incorrectly parsed sentences, clauses and sentence fragments, which the tree-rule processor attempts to match to a rule. Either these fragments have no Theme and therefore no rule is expected, or the sentence is incorrectly parsed (see Section 4.4) and no rule can be expected to match. *Matches MISMATCHed* are all rules that matched, though another rule was expected (*false positives*). The *overall precision* is the percentage of correct matches in the total number of matches (including all *UNEXPECTEDs*).

For this first test we focused on simple Themes. Also, we removed all incorrectly parsed sentences (see Section 4.4 for a discussion of limitations on our approach). This left a set of 68 sentences of the original 85. For the results, see again Table 3.

In every case where a rule was expected, the correct rule was found. The *UNEXPECTED* matches are sentences or sentence fragments that matched a rule that was not expected, because no Theme could be identified for the fragments in the *training mode*.

4.2 Set of sample texts

The second data set on which we tested our approach is composed of 48 sentences from four small texts. These texts are a mixture of academic abstracts and general interest texts. They were chosen for their variety, in order to expose the tree-rule processor to a wide variety of sentence types. In this test, we did not remove incorrect parse trees or complex Themes.

For all four texts, over 83% of the sentences could be classified. A precision of 100.00% was achieved in all cases where a rule was expected. The *sentences with no match* are cases of complex Themes or incorrect parse trees.

4.3 Test corpus

The third set of data is a corpus composed of 700 sentences in 209 academic abstracts from the fields of mechanical engineering, linguistics, and computer science. Here, we were able to classify 89% of the sentences with a precision of 81.74%. Across the whole corpus, we had only 6 *Matches MISMATCHed* in close to 1000 total matches.

4.4 Assessment of results

The results of our tests are a good indication that our approach is viable for automatically identifying Themes, and that the rules we have established so far are a solid foundation on which we can expand our approach. However, there are some problems concerning the syntactic parses and some limitations concerning the Theme identification procedure.

Clearly, the performance of the tree-rule processor depends on the performance of the parser. The more complex a sentence, the more likely it is that the input from the parser is “unclean”. For instance, in example 11, our tree-rule processor will attempt to match a rule for each clause, but the incorrect parse (see Figure 6) produced by the PCFG version of the Stanford parser prevents it from correctly identifying the rule for the inverted clause.

11. *Beyond the main complex is a lovely stream that bubbles under a wooden bridge, and further on are steep stone steps leading to another complex.*

Table 2. Performance on the Test Corpus

Sentences found		68
Sentences with no match		0
Classified sentences		100.0%
Total matches		102
Avg matches per sentence		1.50
Total matches MET	(true positives)	73
Total matches MISMATCHed	(false positives)	0
Total matches UNEXPECTED		29
Overall precision	(with unexpected)	71.57%

Table 3. Performance on sample sentences

Sentences found		48
Sentences with no match		8
Classified sentences		83.33%
Total matches		77
Avg matches per sentence		1.60
Total matches MET	(true positives)	46
Total matches MISMATCHed	(false positives)	0
Total matches UNEXPECTED		31
Overall precision	(with unexpected)	59.74%

Table 4. Performance on Test Set

Sentences found		700
Sentences with no match		75
Classified sentences		89.28%
Total matches		973
Avg matches per sentence		1.39
Total matches MET	(true positives)	855
Total matches MISMATCHed	(false positives)	6
Total matches UNEXPECTED		185
Overall precision	(with unexpected)	81.74%

The Factored model of the parser returns better results in cases of complex sentence structure, as can be seen from the parse of example 11 shown in Figure 7 below. Yet for the majority of simple sentences, the PCFG model provides more consistent results. One solution to this problem would be to adopt a layered approach, where complex sentences are re-parsed using the Factored model.

Another issue is the occurrence of rules matching sentence fragments. This inflates the number of unexpected results, and has a negative effect on the precision of

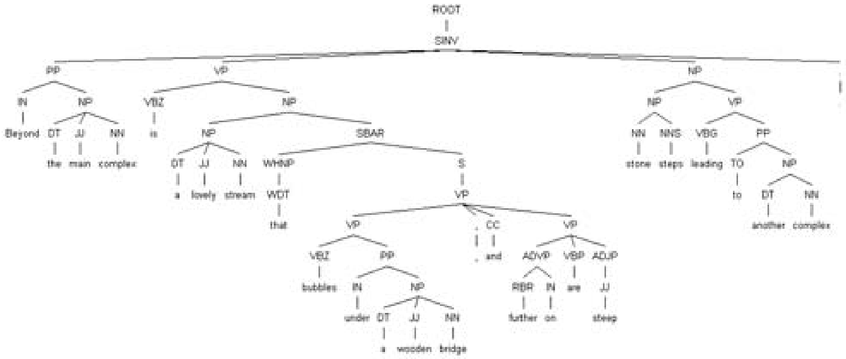

ROOT
SINV
PP VP NP .
IN NP VBZ NP NP VP
Beyond DT JJ NN is NP SBAR NN NNS VBG PP
the main complex DT JJ NN WHNP S stone steps leading TO NP
a lovely stream WDT VP to DT NN
that VP , CC VP another complex
VBZ PP , and ADVP VBP ADJP
bubbles IN NP RBR IN are JJ
under DT JJ NN further on steep
a wooden bridge

Figure 6. Stanford PCFG parse of a sentence containing inversion and coordination

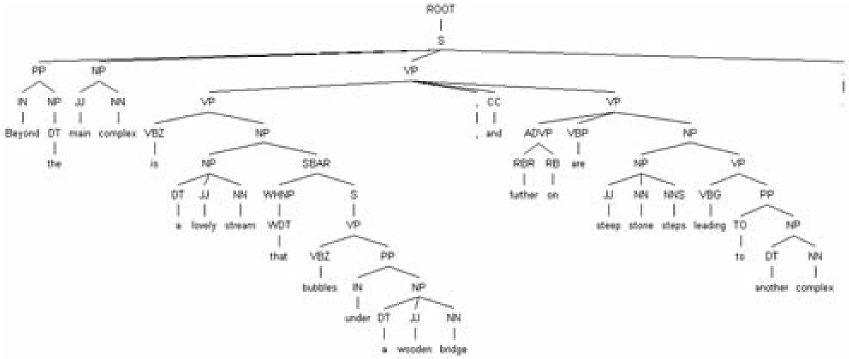

ROOT
S
PP NP VP .
IN NP JJ NN VP , CC VP
Beyond DT main complex VBZ NP , and ADVP VBP NP
the is NP SBAR RBR RB are NP VP
DT JJ NN WHNP S further on JJ NN NNS VBG PP
a lovely stream WDT VP steep stone steps leading TO NP
that VBZ PP to DT NN
bubbles IN NP another complex
under DT JJ NN
a wooden bridge

Figure 7. Stanford Factored parse of a sentence containing inversion and coordination

the tree-rule processor. To counteract this, negative rules can be developed that are more specific to the problematic fragments than the Theme rules. This would label the sentence fragments for later removal from the number of matches and prevent them from being erroneously labelled as a Theme. To test this theory, one negative rule has been introduced to eliminate rules matching *"to" + verb* constructions, for example in *We decided to go to the movies.* In our set of sample sentences from Halliday, this negative rule reduced the number of unexpected matches from 29 to 22. In the test corpus, the number of unexpected matches drops with the inclusion of the negative rule from 185 to 93. This increases the precision of the tree-rule processor on the test corpus by 4.87%. The goal is to continue identifying patterns that could be eliminated using a negative rule in order to increase the precision and reliability of the program.

5 Summary and conclusion

Automatic Theme identification has long been on the agenda for desirable annotation types that have the potential of sparking progress in studies of discourse organization. Recent research efforts in this area clearly show that in order to seriously push corpus-based discourse studies and scalable NLP applications beyond the level of the sentence, annotations at the discourse level are indispensable and preferably attained with minimal manual intervention.

This paper proposes an implementation for Theme identification on the basis of a relatively simple pattern matching algorithm that matches a set of well-defined linguistic rules against a syntactically parsed text corpus. The approach adopted is able to deal with different types of Theme in clauses with all possible mood configurations as well as with all standard simple Theme types described in the literature. Even though we do not achieve 100 % accuracy on free text, the approach adopted already delivers good performance in the identification of simple Theme types.

The current limitations of the approach lie in two areas: firstly, multiple Themes are not covered yet; and secondly, complex sentences produce erroneous parses with the PCFG model of the Stanford parser. We will thus need to expand the rule set for Theme identification and find a way of working around the parser's problems. The observation of the discrepancies between the performance of the PCFG and the Factored parsers on clauses with different levels of complexity requires further testing in order to evaluate the possibilities of a combined application of the two parsers in the hope that this will deliver the most optimal parsing results for our Theme rule interpreter. Finally, an expanded rule set is going to be applied to more text from additional genres and more diverse registers and again evaluated in terms of performance. In terms of applications, we want to analyze patterns of thematic progression on the basis of a Theme-annotated corpus (see e.g., Teich (2006)) as well as explore the possibilities of data mining for detecting differences and commonalities between register-diversified corpora on the basis of Theme information.

References

Buráňová, Eva, Eva Hajičová, and Petr Sgall (2000). Tagging of very large corpora: Topic-focus articulation. In *Proceedings of the 18th conference on Computational Linguistics (Coling)*, volume 1, 139–144, Saarbrücken, Germany.

Eckart, Richard (2006). Towards a modular data model for multi-layer annotated corpora. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 183–190, Sydney, Australia: Association for Computational Linguistics, URL `http://www.aclweb.org/anthology/P/P06/P06-2024`.

Halliday, MAK (1985). *An Introduction to Functional Grammar*. London: Arnold.

Halliday, MAK (2004). *An Introduction to Functional Grammar*. London: Arnold, 3. edition, revised by Matthiessen, C.M.I.M.

Honnibal, Matthew and James R. Curran (2007). Creating a Systemic Functional Grammar Corpus from the Penn Treebank. In *ACL 2007 Workshop on Deep Linguistic Processing*, 89–96, Prague, Czech Republic: Association for Computational Linguistics, URL `http://www.aclweb.org/anthology/W/W07/W07-1212`.

Klein, Dan and Christopher D. Manning (2002). Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS) 2002*, 3–10, Vancouver, British Columbia, Canada.

Lobin, Henning, Maja Bärenfänger, Mirco Hilbert, Harald Lüngen, and Csilla Puskas (2007). Discourse relations and document structure. In Dieter Metzing and Andreas Witt (eds.), *Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology. Serie Text, Speech and Language Technology.*, Dordrecht: Kluwer, to appear.

Lüngen, Harald, Henning Lobin, Maja Bärenfänger, Mirco Hilbert, and Csilla Puskas (2006). Text parsing of a complex genre. In Bob Martens and Milena Dobreva (eds.), *Proceedings of the Conference on Electronic Publishing (ELPUB 2006)*, Bansko, Bulgarien.

Marcu, Daniel (2000). The Rhetorical Parsing of Unrestricted Texts: A Surface-Based Approach. *Computational Linguistics* 26(3):395–448, URL `citeseer.ist.psu.edu/marcu00rhetorical.html`.

Matthiessen, C.M.I.M. (1995). *Lexicogrammatical cartography - English systems*. Tokyo, Taipei, Dallas: International Language Science Publishers.

Poesio, Massimo, Rosemary Stevenson, Barbara di Eugenio, and Janet Hitzeman (2004). Centering: A Parametric Theory and its Instantiations. *Computational Linguistics* 30(3):309–363.

Postolache, Oana, Ivana Kruijff-Korbayová, and Geert-Jan M. Kruijff (2005). Data-Driven Approaches for Information Structure Identification. In *HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 9–16, Morristown, NJ, USA: Association for Computational Linguistics, doi:http://dx.doi.org/10.3115/1220575.1220577.

Stede, M. and S. Heintze (2004). Machine-Assisted Rhetorical Structure Annotation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland.

Teich, Elke (2006). Information Load in Theme and News: an Exploratory Study of Science Texts. In S. Cho and E. Steiner (eds.), *Information Distribution in English Grammar and Discourse and Other Topics in Linguistics*, Frankfurt a. Main: Lang.

Finding canonical forms for historical German text

Bryan Jurish

Abstract. Historical text presents numerous challenges for contemporary natural language processing techniques. In particular, the absence of consistent orthographic conventions in historical text presents difficulties for any technique or system requiring reference to a fixed lexicon accessed by orthographic form. This paper presents two methods for mapping unknown historical text types to one or more synchronically active canonical types: conflation by phonetic form, and conflation by lemma instantiation heuristics. Implementation details and evaluation of both methods are provided for a corpus of historical German verse quotation evidence from the digital edition of the *Deutsches Wörterbuch*.

1 Introduction

Historical text presents numerous challenges for contemporary natural language processing techniques. In particular, the absence of consistent orthographic conventions in historical text presents difficulties for any technique or system requiring reference to a fixed lexicon accessed by orthographic form, such as document indexing systems (e.g. Sokirko 2003), part-of-speech taggers (e.g. Brill 1992; DeRose 1988; Jurish 2003; Schmid 1994), simple word stemmers (e.g. Lovins 1968; Porter 1980), or more sophisticated morphological analyzers (e.g. Geyken and Hanneforth 2006). When adopting historical text into such a system, one of the most important tasks is the discovery of one or more *canonical extant forms* for each word of the input text: synchronically active text types which best represent the historical input form.[1]

The process of collecting variant forms into equivalence classes represented by one or more canonical extant types is commonly referred to as *conflation*, and the equivalence classes themselves are referred to as *conflation sets*. Given a high-coverage analysis function for extant forms, an unknown (historical) form w can then be analyzed as the disjunction of analyses over (the extant members of) its conflation set $[w]$:

$$\text{analyses}(w) \quad := \quad \bigcup_{v \in [w]} \text{analyses}(v)$$

[1] As an anonymous reviewer pointed out, the absence of consistent orthographic conventions is not restricted to corpora of historical text. Various other types of text corpora – including transcriptions of spoken language, corpora containing transcription errors, and corpora for languages with non-standard orthography – might also benefit from a canonicalization strategy such as those presented here.

This paper describes two methods for finding conflation sets in a corpus of *circa* 5.5 million words of historical German verse extracted from quotation evidence in the digital edition of the *Deutsches Wörterbuch* (*DWB*, Bartz et al. 2004), and indexed with the the TAXI document indexing system. The conflation methods were implemented on the entire corpus as a TAXI plug-in module (TAXI/Grimm), and evaluated with respect to coverage by the TAGH morphology.

The rest of this paper is organized as follows: Section 2 describes the first conflation strategy, based on identity of phonetic forms. The second strategy making use of *a priori* assumptions regarding corpus structure and permitting "fuzzy" matching via phonetic edit distance is presented in Section 3. Finally, Section 4 contains a brief summary of the preceding sections and a sketch of the ongoing development process.

2 Conflation by phonetic form

Although the lack of consistent orthographic conventions for middle high German and early new high German texts led to great diversity in surface graphemic forms, we may assume that graphemic forms were constructed to reflect phonetic forms. Under this assumption, together with the assumption that the phonetic system of German is diachronically more stable than the graphematic system, the phonetic form of a word type should provide a better clue to the extant lemma of a historical word than its graphemic form. This insight is the essence of the "conflation by phonetic form" strategy as implemented in the TAXI/Grimm index module.

In order to map graphemic forms to phonetic forms, we may avail ourselves of previous work in the realm of text-to-speech synthesis, a domain in which the discovery of phonetic forms for arbitrary text is a well-known and often-studied problem (cf. Allen et al. 1987; Dutoit 1997; Liberman and Church 1992), the so-called "grapheme-to-phoneme", "grapheme-to-phone", or "letter-to-sound" (LTS) conversion problem. Use of a full-fledged LTS conversion module to estimate phonetic forms provides a more flexible and finer-grained approach to canonicalization by phonetic form than strategies using language-specific phonetically motivated digest codes such as those described in Robertson and Willett (1993). The grapheme-to-phone conversion module in the TAXI/Grimm system uses the LTS rule-set distributed with the IMS German Festival package (Möhler et al. 2001), a German language module for the Festival text-to-speech system (Black and Taylor 1997; Taylor et al. 1998).

2.1 Implementation

As a first step, the IMS German Festival letter-to-sound (LTS) rule-set was adapted to better accommodate both historical and contemporary forms; assumedly at the expense of precision for both historical and contemporary forms. In particular, the following changes were made:

1. By default, the grapheme “h” is ignored (considered silent).

2. A single additional rule maps the grapheme sequence “sz” to voiceless /*s*/.

3. Vowel-length estimates output by the IMS German rule-set are ignored; thus /*e*/ and /*eː*/ are both mapped to the canonical phonetic form /*e*/.

4. Schwas (/ə/) predicted by the IMS German rule-set are replaced by /*e*/ in the canonical phonetic form.

5. Adjacent occurrences of any single vowel predicted by the IMS German rule-set are replaced by a single occurrence, thus /*aa*/, /*aaa*/, and /*aaaa*/ are all mapped to /*a*/.

The adapted rule-set was converted to a *deterministic finite state transducer* (Aho and Ullman 1972; Roche and Schabes 1997) using the GFSM finite state machine utility library. Formally, the finite state transducer (FST) used by the TAXI/Grimm LTS module is defined as the machine M_{LTS} arising from the composition of two *Aho-Corasick pattern matchers* (Aho and Corasick 1975) M_L, M_R and an additional *output filter* M_O:

$$M_{LTS} = (M_L \circ M_R \circ M_O) : \mathscr{A}_g^* \to \mathscr{A}_p^* \tag{3.1}$$

where $\mathscr{A}_g$ is the finite *grapheme alphabet* and $\mathscr{A}_p$ is the finite *phone alphabet*. To define the individual component machines, let R be the (IMS German) Festival LTS rule-set source, a finite set of rules of the form $(\alpha[\beta]\gamma \to \pi) \in \mathscr{A}_g^* \times \mathscr{A}_g^+ \times \mathscr{A}_g^* \times \mathscr{A}_p^*$, read as: the *source grapheme string* β is to be mapped to the *target phonetic string* π if β occurs with *left graphemic context* α and *right graphemic context* γ; let $\prec$ be a linear *precedence order* on R which prevents multiple rules from applying to the same source substring (only the $\prec$-minimal rule is applied at each source position, proceeding from left to right); for a nonempty rule subset $S \subseteq R$, let $(\alpha_S[\beta_S]\gamma_S \to \pi_S) = \min_\prec S$; let $\mathrm{AhoCorasick}(P) : \mathscr{A}^* \to \wp(P)^*$ be the Aho-Corasick pattern matcher for a set P of string patterns from a finite alphabet $\mathscr{A}$; let $|\cdot|$ denote string length or set cardinality, depending on context; let $\mathrm{reverse}(\cdot)$ denote the transducer reversal operation, and let $\mathrm{Concat}(\cdots)$ denote the string concatenation operation, then:

$$\begin{aligned} M_L &\approx \mathrm{AhoCorasick}\left(\{\alpha : (\alpha[\beta]\gamma \to \pi) \in R\}\right) \\ &: \mathscr{A}_g^* \to (\mathscr{A}_g \times \wp(R))^* \\ &: w \mapsto \mathrm{Concat}_{i=0}^{|w|} \left\langle w_i, \left\{(\alpha[\beta]\gamma \to \pi) \in R \mid w_{(i-|\alpha|)..i} = \alpha\right\}\right\rangle \end{aligned} \tag{3.2}$$

$$\begin{aligned} M_R &\approx \mathrm{reverse}\left(\mathrm{AhoCorasick}\left(\{(\beta\gamma)^{-1} : (\alpha[\beta]\gamma \to \pi) \in R\}\right)\right) \\ &: (\mathscr{A}_g \times \wp(R))^* \to \wp(R)^* \\ &: \langle w_i, S_i \rangle_I \mapsto \mathrm{Concat}_{i \in I}\left(S_{i-1} \cap \left\{(\alpha[\beta]\gamma \to \pi) \in R : w_{i..(i+|\beta\gamma|)} = \beta\gamma\right\}\right) \end{aligned} \tag{3.3}$$

A similar construction also using a pair of Aho-Corasick pattern matchers (analogous to M_L and M_R) is employed by Laporte (1997) for compiling a single bimachine from a set of conflict-free hand-written phonetic conversion rules. Since `festival` LTS rule-sets are not conflict-free, Laporte's technique cannot be applied directly here, and the choice of which rule to apply must be delayed until application of the filter transducer M_O:

$$\begin{aligned} M_O &\approx \left(\bigcup_{S \in \wp(R)} \left[(S : \pi_S)\, (\wp(R) : \varepsilon)^{|\beta_S|-1} \right] \right)^* \\ &: \wp(R)^* \to \mathscr{A}_p^* \end{aligned} \tag{3.4}$$

In the interest of efficiency, the rule subsets $S \in \wp(R)$ on the lower tape of the filter transducer M_O can be restricted to those which actually occur on the upper tape of the right-context transducer M_R: such a restriction represents a considerable efficiency gain with respect to the "brute force" power-set construction given in Equation 3.4. Figure 1 shows an example of how the various machine components work together to map the graphemic form "sache" to the phonetic form /*zaxə*/.

Finally, phonetic forms are used to conflate graphemic variants $w \in \mathscr{A}$ as equivalence classes $[w]_{\mathrm{pho}}$ with respect to the *phonetic equivalence relation* $\equiv_{\mathrm{pho}}$ on the corpus word-type alphabet $\mathscr{A} \subset \mathscr{A}_g^*$:

$$w \equiv_{\mathrm{pho}} v \quad :\Leftrightarrow \quad M_{LTS}(w) = M_{LTS}(v) \tag{3.5}$$

$$[w]_{\mathrm{pho}} \quad = \quad \{v \in \mathscr{A} : w \equiv_{\mathrm{pho}} v\} \tag{3.6}$$

Note that the equivalence class generating function $[\cdot]_{\mathrm{pho}} : \mathscr{A} \to \wp(\mathscr{A})$ can itself be characterized as a finite state transducer, defined as the composition of the LTS transducer with its inverse, and restricted to the alphabet $\mathscr{A}$ of actually occurring corpus word-types:

$$[\cdot]_{\mathrm{pho}} \quad := \quad M_{LTS} \circ M_{LTS}^{-1} \circ \mathrm{Id}\,(\mathscr{A}) \tag{3.7}$$

Input	#	s	a	c	h	e	#
$\mathbf{M_L}$ $\longrightarrow$	$\emptyset$	$\left\{\begin{array}{ll}[a]ch & \to a\\ [a] & \to aː,\\ [c] & \to k,\\ [e] & \to ə,\\ \#[s]a & \to z,\\ [s] & \to s\end{array}\right\}$	$\left\{\begin{array}{ll}[a]ch & \to a,\\ [a] & \to aː,\\ [c] & \to k,\\ [e] & \to ə,\\ [s] & \to s\end{array}\right\}$	$\left\{\begin{array}{ll}[\,a\,]ch & \to a,\\ [\,a\,] & \to aː,\\ a[ch] & \to x,\\ [\,c\,] & \to k,\\ [\,e\,] & \to ə,\\ [\,s\,] & \to s\end{array}\right\}$	$\emptyset$	$\left\{\begin{array}{ll}[a]ch & \to a,\\ [a] & \to aː,\\ [c] & \to k,\\ [e] & \to ə,\\ [s] & \to s\end{array}\right\}$	$\emptyset$
$\mathbf{M_R}$ $\longleftarrow$	$\emptyset$	$\left\{\begin{array}{l}\#[s]a\to z,\\ [s]\ \to s\end{array}\right\}$	$\left\{\begin{array}{l}[a]ch\to a,\\ [a]\ \to aː\end{array}\right\}$	$\left\{\begin{array}{l}a[ch]\to x,\\ [\,c\,]\to k\end{array}\right\}$	$\emptyset$	$\left\{\ [e]\to ə\ \right\}$	$\emptyset$
$\mathbf{M_O}$ $\longrightarrow$	ε	**z**	**a**	**x**	ε	**ə**	ε

Figure 1. Example Letter-to-Sound Transduction from "sache" to/*zaxə*/. Here, italic "ε" indicates the empty (phonetic) string.

2.2 Performance

Table 1. Performance results for LTS FST *vs.* direct communication with a `festival` process

LTS Method	**Throughput (tok/sec)**	**Relative**
`festival` (TCP)	28.53	−4875.57 %
`festival` (pipe)	1391.45	± 0.00 %
FST (`libgfsm`)	9124.69	+ 555.77 %

A finite state LTS transducer M_{LTS} was compiled from the 396 rules of the adapted IMS German Festival rule-set using the procedure sketched above. The resulting transducer contained 131,440 arcs and 1,037 states, of which 292 were final states. The compilation lasted less than 30 seconds on a workstation with a 1.8GHz dual-core processor. Performance results for the transducer representation of the LTS rule-set and for two methods using `festival` directly are given in Table 1. As expected, the transducer implementation was considerably faster than either of the methods communicating directly with a `festival` process.

Table 2. Some words conflated by identity of phonetic form

Extant Form w	**Phonetic Equivalence Class** $[w]_{\text{pho}}$
fröhlich	*frölich, fröhlich, vrœlich, frœlich, froͤlich, froͤhlich, vrölich, frölig, …*
Herzenleid	*hertzenleid, herzenleid, herzenleit, hertzenleyd, hertzenleidt,* *herzenlaid, hertzenlaid, hertzenlaidt, hertzenlaydt, herzenleyd, …*
Hochzeit	*hochtzeit, hochzeit, hochzeyt, hochzît, hôchzît, hochzeid, …*
Schäfer	*schäfer, schäffer, scheffer, scheppher, schepher, schaͤfer, schähffer, …*

2.3 Coverage

The phonetic conflation strategy was tested on the full corpus of the verse quotation evidence extracted from the DWB, consisting of 6,581,501 tokens of 322,271 distinct graphemic word types. A preprocessing stage removed punctuation marks, numerals, and known foreign-language material from the corpus. Additionally, a rule-based graphemic normalization filter was applied which maps UTF-8 characters not occurring in contemporary German orthography onto the ISO-8859-1 (Latin-1) character set (e.g. *œ*, *oͤ*, and *ô* are mapped to *oe*, *ö*, and *o*, respectively). After preprocessing and filtering, the corpus contained 5,491,982 tokens of 318,383 distinct ISO-8859-1 encoded graphemic types.

Of these 318,383 Latin-1 word types occurring in the corpus, 135,070 (42.42%) were known to the TAGH morphology (Geyken and Hanneforth 2006), representing a total coverage of 4,596,962 tokens (83.70%). By conflating those word types which share a phonetic form according to the LTS module, coverage was extended to a total of 173,877 (54.61%) types, representing 5,028,999 tokens (91.57%). Thus, conflation by phonetic form can be seen to provide a reduction of 21.17% in type-wise coverage errors, and of 48.27% in token-wise coverage errors. Some examples of word types conflated by the phonetic canonicalization strategy are given in Table 2.

3 Conflation by lemma instantiation heuristics

Despite its encouragingly high coverage, conflation by identity of phonetic form is in many cases too strict a criterion for lemma-based canonicalization – many word pairs which intuitively should be considered instances of the same lemma are assigned to distinct phonetic equivalence classes. Examples of such desired conflations undiscovered by the phonetic conflation strategy include the pairs *(abbrechen, abprechen)*, *(geschickt, geschicket)*, *(gut, guot)*, *(Licht, liecht)*, *(Teufel, tiuvel)*, *(umgehen, umbgehn)*, *(voll, vol)*, and *(wollen, wolln)*. In an attempt to address these shortcomings of the phonetic conflation method, additional conflation heuristics were developed which make use of the dictionary structure of the TAXI/Grimm corpus in order to estimate and maximize a *lemma instantiation likelihood* function.

3.1 Implementation

The TAXI/Grimm corpus is comprised of *verse quotation evidence* drawn from a dictionary corpus (Bartz et al. 2004). It is plausible to assume that each of the quotations occurring in an article for a particular dictionary lemma contain some variant of that lemma – otherwise there would not be much sense including the quotation as "evidence" for the lemma in question.

Working from this assumption that each quotation contains at least one variant of the dictionary lemma for which that quotation appears as evidence, a lemma instantiation conflation heuristic has been developed which does not require strict identity of phonetic forms – instead, *string edit distance* (Levenshtein 1966; Navarro 2001; Wagner and Fischer 1974) on phonetic forms is used to estimate similarity between each word in the corpus and each of the dictionary lemmata under which it occurs. Further, inspired by previous work in unsupervised approximation of semantics and morphology (Baroni et al. 2002; Church and Hanks 1990; Yarowsky and Wicentowski 2000), *pointwise mutual information* (Cover and Thomas 1991; Manning and Schütze 1999; McGill 1955) between dictionary lemmata and their candidate instances is employed to detect and filter out "chance" similarities between rare lemmata and high-frequency words.

Formally, the lemma instantiation heuristics attempt to determine for each quotation q which phonetic type i occurring in q best instantiates the dictionary lemma ℓ associated with the article containing q. For $\mathscr{A}$ the set of all word types occurring in the corpus, $\mathscr{L} \subseteq \mathscr{A}$ the set of all dictionary lemmata, and $\mathscr{Q} \subseteq \wp(\mathscr{A}^*)$ the set of all quotations:

$$\begin{aligned} \text{bestInstance}(\cdot) \quad &: \quad \mathscr{Q} \to \mathscr{A} \\ &: \quad q \mapsto \arg\max_{w \in q} \mathrm{L}(M_{LTS}(w), M_{LTS}(\text{lemma}(q))) \end{aligned} \tag{3.8}$$

where the probabilities $\mathrm{P}(\ell, i), \mathrm{P}(\ell)$, and $\mathrm{P}(i)$ used to compute pointwise mutual information are first instantiated by their maximum likelihood estimates over the entire corpus:

$$\mathrm{P}(\ell,i) = \frac{\sum_{w_i \in M_{LTS}^{-1}(i)} \sum_{w_\ell \in M_{LTS}^{-1}(\ell)} \mathrm{f}(Token = w_i, Lemma = w_\ell)}{|Corpus|} \tag{3.9}$$

$$\mathrm{P}(\ell) = \sum_i \mathrm{P}(\ell,i) \tag{3.10}$$

$$\mathrm{P}(i) = \sum_\ell \mathrm{P}(\ell,i) \tag{3.11}$$

Raw bit-length pointwise mutual information values $\tilde{\mathrm{I}}(\ell,i)$ are computed and normalized to the unit interval $[0,1]$ for each lemma and candidate instance, defining $\tilde{\mathrm{I}}(i|\ell)$ and $\tilde{\mathrm{I}}(\ell|i)$ respectively:

$$\tilde{\mathrm{I}}(\ell,i) = \log_2 \frac{\mathrm{P}(\ell,i)}{\mathrm{P}(\ell)\mathrm{P}(i)} \tag{3.12}$$

$$\tilde{\mathrm{I}}(i|\ell) = \frac{\tilde{\mathrm{I}}(\ell,i) - \min \tilde{\mathrm{I}}(\ell,\mathscr{A})}{\max \tilde{\mathrm{I}}(\ell,\mathscr{A}) - \min \tilde{\mathrm{I}}(\ell,\mathscr{A})} \tag{3.13}$$

$$\tilde{\mathrm{I}}(\ell|i) = \frac{\tilde{\mathrm{I}}(\ell,i) - \min \tilde{\mathrm{I}}(\mathscr{L},i)}{\max \tilde{\mathrm{I}}(\mathscr{L},i) - \min \tilde{\mathrm{I}}(\mathscr{L},i)} \tag{3.14}$$

The user-specified function $d_{\mathrm{max}}(\ell,i)$ serves a dual purpose: first as a normalization factor for the fuzzy phonetic similarity estimate $\mathrm{sim}(\ell,i)$, and second as a cutoff threshold for absolute phonetic edit distances $\mathrm{d}_{\mathrm{edit}}(\ell,i)$, blocking instantiation hypotheses when phonetic dissimilarity grows “too large”:

$$d_{\mathrm{max}}(\ell,i) = \min\{|\ell|,|i|\} - 1 \tag{3.15}$$

The lemma instantiation likelihood function $\mathrm{L}(i,\ell)$ is defined as the product of the normalized phonetic similarity and the arithmetic average component-normalized mutual information score:

$$\mathrm{sim}(\ell,i) = \begin{cases} \frac{d_{\mathrm{max}}(\ell,i) - \mathrm{d}_{\mathrm{edit}}(\ell,i)}{d_{\mathrm{max}}(\ell,i)} & \text{if } \mathrm{d}_{\mathrm{edit}}(\ell,i) \leq d_{\mathrm{max}}(\ell,i) \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

$$\mathrm{L}(i,\ell) = \frac{\mathrm{sim}(\ell,i) \times (\tilde{\mathrm{I}}(\ell|i) + \tilde{\mathrm{I}}(i|\ell))}{2} \tag{3.17}$$

Finally, the edit-distance lemma instantiation heuristic conflates those word pairs which share either a phonetic form or appear as best instances of some common dictionary lemma:[2]

$$w \equiv_{\text{li}} v \quad :\Leftrightarrow \quad \begin{array}{l}(w \equiv_{\text{pho}} v) \text{ or} \\ (\text{bestInstance}^{-1}(w) \cap \text{bestInstance}^{-1}(v) \neq \emptyset)\end{array} \tag{3.18}$$

3.2 Performance

A major advantage of this approach arises from the relatively small number of edit distance comparisons which must be performed. Since the Wagner-Fischer algorithm (Wagner and Fischer 1974) used to compute phonetic edit distances has quadratic running time, $\mathbf{O}(\mathrm{d}_{\text{edit}}(w,v)) = \mathbf{O}(|w||v|)$, the number of edit distance comparisons comprises the bulk of the heuristic's running time, and should be kept as small as possible. Restricting the comparisons to those pairs (ℓ, i) of dictionary lemmata and phonetic types occurring in quotation evidence for those lemmata requires that approximately 3.38 million comparisons be made during analysis of the entire TAXI/Grimm quotation corpus. If instead every possible unordered pair of phonetic types were to be compared – as required by some morphology induction techniques – a total of *circa* 340 billion comparisons would be required, over ten thousand times as many! With restriction of comparisons to dictionary lemmata, the heuristic analysis completes in 28 minutes on a 1.8GHz dual-core processor workstation, which corresponds to a projected running time of about 5.35 years for a method comparing all unordered word pairs, which is clearly unacceptable.

3.3 Coverage

Using the verse quotation evidence corpus described above in Section 2.3, the lemma instantiation conflation heuristics discovered conflations with extant forms known to the TAGH morphology for 29,248 additional word types not discovered by phonetic conflation, including all of the example word pairs given in the introduction to this section. Additionally, 9,415 word types were identified as "best instances" for DWB lemmata unknown to the TAGH morphology. Together with phonetic conflation, the lemma instantiation heuristics achieve a total coverage of 212,540 types (66.76%), representing 5,185,858 tokens (94.43%). Thus, the lemma instantiation heuristic conflation method provides a reduction of 26.76% in type-wise coverage errors and of 33.88% in token-wise coverage errors with respect to the phonetic

[2] Note that $\equiv_{\text{li}}$ is not an equivalence relation in the strict sense, since although it is reflexive and symmetric, it is not transitive.

identity conflation method alone, resulting in a total reduction of 42.26% in type-wise coverage errors and of 65.80% in token-wise coverage errors with respect to the literal TAGH morphology.

4 Summary and outlook

Two strategies were presented for discovering synchronically active canonical forms for unknown historical text forms. Together, the two methods achieve TAGH morphological analyses for 94.43% of tokens, reducing the number of unknown tokens by 65.8% in a corpus of *circa* 5.5 million words of historical German verse. In the interest of generalizing these strategies to arbitrary input texts, a robust system for lazy online best-path lookup operations in weighted finite state transducer cascades (such as phonetic equivalence classes or best-alignments with a target language in the form of a finite state acceptor) is currently under development.

While the high coverage rate of the conflation strategies presented here is encouraging, a number of important questions remain. Chief among these is the question of the canonicalization strategies' reliability: how many of the discovered extant "canonical" forms are in fact morphologically related to the source forms? Conversely, were all valid canonical forms for each covered source word indeed found, or were some missed? A small gold standard test corpus is currently under construction which should enable quantitative answers to these questions in terms of the information retrieval notions of *precision* and *recall*.

References

Aho, Alfred V. and Margaret J. Corasick (1975). Efficient String Matching: an Aid to Bibliographic Search. *Commun. ACM* 18(6):333–340, doi:http://doi.acm.org/10.1145/360825.360855.

Aho, Alfred V. and Jefffrey D. Ullman (1972). *The Theory of Parsing, Translation and Compiling*. Englewood Cliffs, N.J.: Prentice-Hall.

Allen, J., S. Hunnicut, and D. Klatt (1987). *From Text to Speech: the MITalk System*. Cambridge University Press.

Baroni, Marco, Johannes Matiasek, and Harald Trost (2002). Unsupervised Discovery of Morphologically Related Words Based on Orthographic and Semantic Similarity. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-2002*, 48–57.

Bartz, Hans-Werner, Thomas Burch, Ruth Christmann, Kurt Gärtner, Vera Hildenbrandt, Thomas Schares, and Klaudia Wegge (eds.) (2004). *Der Digitale Grimm. Deutsches Wörterbuch von Jacob und Wilhelm Grimm*. Frankfurt am Main: Zweitausendeins, URL `http://www.dwb.uni-trier.de`.

Black, Alan W. and Paul Taylor (1997). Festival Speech Synthesis System: System Documentation. Technical Report HCRC/TR-83, University of Edinburgh, Centre for Speech Technology Research, URL `http://www.cstr.ed.ac.uk/projects/festival`.

Brill, Eric (1992). A Simple Rule-Based Part-of-Speech Tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, 152–155, Trento, Italy.

Church, Kenneth Ward and Patrick Hanks (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics* 16(1):22–29.

Cover, Thomas M. and Joy A. Thomas (1991). *Elements of Information Theory*. New York: John Wiley & Sons.

DeRose, Stephen (1988). Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics* 14(1):31–39.

Dutoit, Thierry (1997). *An Introduction to Text-to-Speech Synthesis*. Dordrecht: Kluwer.

Geyken, Alexander and Thomas Hanneforth (2006). TAGH: A Complete Morphology for German based on Weighted Finite State Automata. In *Finite State Methods and Natural Language Processing. 5th International Workshop, FSMNLP 2005, Helsinki, Finland, September 1-2, 2005. Revised Papers*, volume 4002, 55–66, Springer, doi:http://dx.doi.org/10.1007/11780885_7.

Jurish, Bryan (2003). A hybrid approach to Part-of-Speech Tagging. Technical report, Project "Kollokationen im Wörterbuch", Berlin-Brandenburg Academy of Sciences, Berlin, URL `http://www.ling.uni-potsdam.de/~jurish/pubs/dwdst-report.pdf`.

Laporte, Éric (1997). Rational Transductions for Phonetic Conversion and Phonology. In Roche and Schabes (1997).

Levenshtein, Vladimir I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady* 10(1966):707–710.

Liberman, M. J. and Kenneth Ward Church (1992). Text Analysis and Word Pronunciation in Text-to-Speech Synthesis. In Sadaoki Furui and M. Mohan Sondhi (eds.), *Advances in Speech Signal Processing*, New York: Dekker.

Lovins, Julie Beth (1968). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics* 11:22–31.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

McGill, W. J. (1955). Multivariate Information Transmission. *IEEE Trans. Inf. Theory* 4(4):93–111.

Möhler, Gregor, Antje Schweitzer, and Mark Breitenbücher (2001). *IMS German Festival Manual, Version 1.2*. Institute for Natural Language Processing, University of Stuttgart, URL `http://www.ims.uni-stuttgart.de/phonetik/synthesis`.

Navarro, Gonzalo (2001). A Guided Tour to Approximate String Matching. *ACM Computing Surveys* 33(1):31–88.

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program* 14(3):130–137.

Robertson, Alexander M. and Peter Willett (1993). A Comparison of Spelling-Correction Methods for the Identification of Word Forms in Historical Text Databases. *Literary and Linguistic Computing* 8(3):143–152.

Roche, Emmanuel and Yves Schabes (eds.) (1997). *Finite–State Language Processing*. Cambridege, Massachusetts: MIT Press.

Schmid, Helmut (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, 44–49, Manchester, UK.

Sokirko, Alexey (2003). A Technical Overview of DWDS/Dialing Concordance. Talk delivered at the Meeting *Computational Linguistics and Intellectual Technologies*, Protvino, Russia, URL `http://www.aot.ru/docs/OverviewOfConcordance.htm`.

Taylor, Paul, Alan W. Black, and Richard J. Caley (1998). The Architecture of the the Festival Speech Synthesis System. In *Third International Workshop on Speech Synthesis, Sydney, Australia, November, 1998*.

Wagner, Robert A. and Michael J. Fischer (1974). The String-to-String Correction Problem. *Journal of the ACM* 21(1):168–173.

Yarowsky, David and Richard Wicentowski (2000). Minimally Supervised Morphological Analysis by Multimodal Alignment. In K. Vijay-Shanker and Chang-Ning Huang (eds.), *Proceedings of the 38th Meeting of the Association for Computational Linguistics*, 207–216, Hong Kong.

Computing distance and relatedness of medieval text variants from German

Stefanie Dipper and Bettina Schrader

Abstract. In this paper, we explore several ways of computing similarity between medieval text variants from German. In comparing these texts, we apply methods from word and sentence alignment and compute cosine similarity based on character and part-of-speech ngrams. The resulting similarity (or distance) scores are visualized by phylogenetic trees; the methods correctly reproduce the well-known distinction between Middle and Upper German ("Mitteldeutsch" vs. "Oberdeutsch").

1 Introduction[1]

Research on historical languages and their relationships has long been exclusively in the realm of comparative linguistics. For instance, since the early 19th century, Indo-European linguistics has dealt with the reconstruction of language families and language evolution in Europe. Dialects have received similar attention, as they tend to be more conservative than the standard language and often preserve properties of earlier language stages that have been abandoned by the standard language. Hence, historical as well as dialectal data can provide hints on language evolution.

Traditionally, historical linguistic research is mainly based on morphological, phonetic and phonological properties. The relationships between the Indo-European languages e.g. have been established on the base of shared inflectional properties. Furthermore, sound changes (e.g. the first and second Germanic consonant shift) have been used to draw even finer distinctions. Similarly, dialect classification mainly depends on phonetic-phonological features and rarely takes syntactic properties into account.

Language comparison in this spirit is based on specific language data: for sound-based comparison, lists of parallel words in different languages or language stages are usually used, such as the Swadesh list (Swadesh 1955). Comparisons based on (morpho-) syntactic properties usually use lists of syntactic features whose language-specific values are compared with each another.

More recently people have begun to apply clustering algorithms to such data to compute relationships between dialects (e.g. Nerbonne et al. 1996; Spruit 2006),

[1] We thank the anonymous reviewers for their suggestions and many helpful comments.

and to apply phylogenetic clustering algorithms from bioinformatics to derive language family trees in the vein of Indo-European linguistics (e.g. Gray and Atkinson 2003). Phylogenetic algorithms are normally fed with "parallel" DNA sequences of different species and compute (minimal) trees of modification events that relate the species. When applying the algorithms to language data, researchers assume that DNA sequences and language strings behave similar with regard to evolution. Despite commonalities such as diversification or extinction of species and languages, however, there are also differences. For instance, language *contact* is assumed to play an important role in language evolution.

As an alternative to lists of words or features, *corpora* can be used as the base of comparison, and there has been a long-standing tradition of applications in computational linguistics that measure similarities between texts, like authorship attribution (Mosteller and Wallace 1964), and information retrieval and text classification in general. In such an application, a set of features, like word frequency lists or part-of-speech ngrams, is extracted from a (usually large) collection of reference texts, and subsequently used as base of comparison whenever a new text is being classified.

In our paper, we are comparing historical dialect data using such corpus-based methods. However, as historical or dialectal data typically lack (electronically-available) corpora, we are doing without a large collection of reference data. Instead, we base our study on *parallel* corpora, i.e. text variants that share the same underlying "source" text. Hence, quantitative methods can be applied sensibly even to small amounts of data.

In a similar vein, Lüdeling (2006) reports on a study of a (tiny) parallel corpus of the Lord's Prayer in five language stages and dialects of German: In this study, Lüdeling (2006) enriches the manually-aligned corpus with phonetic and syntactic annotations and uses these annotations as the basis of comparison.

We also explore several ways of computing the similarity between medieval German texts. As phonetic annotations along the lines of Lüdeling (2006) require profound knowledge, sound intuitions and a lot of manual work, we decided to compare "writing dialects" ("Schreibdialekte") instead. Just as (a series of) phonetic similarities do not occur by chance, (a series of) spelling similarities can also be used as indications of similarity. In contrast to the study by Lüdeling (2006), we do not use data that has been (manually) pre-aligned. We use an automatic word alignment procedure, and compute similarity based on the alignment results, and character and part-of-speech ngrams. The resulting similarity measures are used to generate phylogenetic similarity trees. Our work presents a pilot study of a much larger project and, hence, is restricted to five tiny texts, each consisting of around 230 words.

The paper is organized as follows. In Sec. 2, we describe the corpus. Sec. 3 and Sec. 4 present the computation of similarity scores based on word alignment and ngram models, respectively. Sec. 5 presents the conclusion.

Table 1. The five texts of our pilot study and their dialects, along with their larger dialect family (MG: *Middle German*, UG: *Upper German*) and date of origin (century).

1.	OMD (“Ostmitteldeutsch”): from Thuringia	MG, 15th
2.	NURN (“Nürnberg”): from the city of Nürnberg (Bavarian, but close to the Franconian border)	UG, 14th
3.	REG (“Regensburg”): from the region of Regensburg (Bavarian)	UG, 15th
4.	AUG (“Augsburg”): from the region of Augsburg (Bavarian, but close to the Swabian border)	UG, 15th
5.	SCHW (“Schwäbisch”): from the region around Kehl (Swabian, but shows alemanic properties)	UG, 15th

2 The corpus

The texts we used in our pilot study ultimately go back to a Latin source, “Interrogatio Sancti Anselmi de Passione Domini” (‘Questions by Saint Anselm about the Lord’s Passion’). The text consists of a collection of questions posed by Saint Anselm of Canterbury and answered by Saint Mary. In the 14th–16th centuries, this text has been written up in various German dialects (from Upper, Middle, and Low German), and transformed into long and short prose and lyric versions. In total, there are more than 50 manuscripts and prints, which makes the text an exceptionally broadly-documented resource.[2]

The basis of our comparisons are ASCII-transcriptions of the manuscripts, kindly provided to us by the “Altgermanistik” group at Ruhr-Universität Bochum. The transcriptions render the manuscripts as original as possible, so that virtually no interpretation is involved in the transcription process. For our pilot study, we selected five texts from different regions, dialects, and times (see Table 1), and extracted two of Anselm’s questions; average length of the extracts is 234.4 words. According to their classifications as *Middle* vs. *Upper German*, one would expect high similarity scores for Texts 2–5 as opposed to Text 1. Texts 2–4 are Bavarian texts, so they should form a cluster as well; however, as noted in Table 1, only Text 3 is a “pure” Bavarian text.

The texts indeed represent parallel texts, e.g., (manually) aligning corresponding words is straightforward, see below the beginnings of Saint Mary’s answer to the first question, according to the five texts.[3]

[2] A comparable parallel corpus is “The 24 Elders” by Otto of Passau, which is documented by more than 120 manuscripts (not electronically available). Besch (1967) uses 68 of these for a detailed (manual) comparison of their dialects. The comparison focuses on the question which of the dialectal form variants made it into modern German.

[3] Rough translation: ‘as my dear child had eaten the Last Supper together with his disciples before his martyrdom’. The use of special characters in the ASCII-based transcriptions are explained below.

1. OMD Do mein lieber Son jhe$u$z Da$z nachtmal mit $ienen jüngern am heiligen grün dorn$tage ge$$en hatte
2. NURN Do mein kint mit $einen iungernn het ge e$$en vor $einer marter das iüng$t e$$en
3. REG da mei\- libis chind wol gee$$enn mit $eine\- Junger\- vor $einer marter daz le$t mal
4. AUG Do meín chind híet geezzen. mít $eínen Jungern. vor $eíner marter daz íungí$t mal.
5. SCHW Do min kint hatte gezen daz ivnge$te maz mit sinen ivng’n vor sin’ mart’.

The example sentences exhibit interesting peculiarities, e.g. with respect to the vocabulary: in four of the five texts, Mary uses *Kind* ‘child’ to refer to Jesus, in one text only she uses *Sohn* ‘son’; the Last Supper is called *Nachtmahl, jüngstes Essen, letztes/jüngstes Mahl*. The spelling variations *chind/kint* probably reflect phonetic differences; on the other hand, the variations *gee$$en, geezzen* could be solely due to different writing conventions: rendered more closely to the original manuscripts, *$* would look like medial long *ſ*, *z* represents the so-called “tailed z”, which looks like ʒ. The combinations *ſſ* vs. *ʒʒ* do not necessarily involve a phonetic difference: *ſſ* is usually interpreted as a writing variant replacing former *ʒʒ*. A similar case is *u/v* alternation, as in *Jungern/ivngern* ‘disciples’. Capitalization at this time is used rather arbitrarily: capitals sometimes mark the sentence beginnings or highlight certain prominent nouns (or parts thereof) like *Mutter gottes* ‘Mother of God’ (not consistently throughout the texts, though).

Variations of the auxiliary ‘had’ (*hatte, het, híet*) and the participle ‘eaten’ (*gessen* vs. *ge-essen*) presumably indicate (phonetic-)inflectional differences. With regard to syntax, one can observe that Text 1 (OMD) is most similar to modern standard German: it shows the order V>AUX (as is also usual in subordinate clauses in modern German), and all verb complements precede the head verb. In contrast, NURN, AUG and SCHW show AUX>V (the auxiliary is missing in REG), and they either extrapose all (non-subject) complements to the right (REG, AUG, SCHW) or just one complement appears preverbal (NURN).

These differences can be due to various reasons: they either reflect “truly” linguistic differences (such as the phonetic, inflectional, and syntactic variations), or they can be due to different spelling or writing conventions (*ʒʒ* vs. *ſſ*; capitalization). A certain amount of the text properties, however, is not genuine but “inherited”: these properties are due to “translationese” that occurred when one author copied, or even translated, the text from one dialect into the other. In our current study, we have not yet addressed this issue.

Finally, we pre-processed and slightly normalized the texts:

Punctuation: all punctuation marks are deleted (only AUG and SCHW show systematic use of punctuation marks).

Capitalization: all letters are converted to lower case.

Superscripts: superscripts as in $l\overset{i}{v}te$ are rendered as *lv\ite* in the transcriptions. In general, such letter combinations can encode diphthongs or monophthongs (similar to *Umlaut* marking). To ease further processing, we replaced these combinations by the equivalent diphthong letters, e.g. *lvite*.

Abbreviations: all notational abbreviations are spelt out.[4] These include the superscribed horizontal bar ("Nasalstrich") as in *mei\-* (REG), which we replace by *mein*. A superscribed hook ("er-Kürzung") abbreviates *-er*, as in *mart'*, which is replaced by *marter* (SCHW). A frequently-used abbreviation is *vn\-* for the conjunction *und/unde* 'and', which we also eliminated.[5]

3 Similarity based on word alignment

As a means to directly compare the vocabularies and word forms of the five dialects, we aligned them both on the sentence and word level. First, we aligned all ten language pairs (OMD↔NURN, OMD↔REG, etc.) manually on the sentence level. The word alignment has been created automatically using Levenshtein Distance, where deletions, insertions and substitutions of characters are equally punished, and a best-first search computes a full alignment path. Thus, the similarity scores of the aligned word pairs directly reflect their graphemic similarity, and can be analyzed along these lines.[6]

3.1 Quantitative data analysis

The common statistic characteristics of the 10 language pairs show that all dialects appear to be highly similar on the graphemic and lexical level (see Table 2): in all

[4] Of course, the specific use of abbreviations is also part of a writing dialect. We opted for using full forms to support automatic alignment.

[5] In order to know which of the alternative full forms (*und, unde, vnd, vnde*) would have to replace the abbreviated form, we counted all full occurrences in the texts: OMD uses 43x *vnde* and 1x *vnd*, NURN 58x *vnd* and no other variant, etc. Unfortunately, SCHW only uses the abbreviated form, so we had to introduce an artificial "neutral" letter for the full forms in SCHW: *vnd@*.

[6] The extracts were too small to use an off-the-shelf statistical alignment tool like GIZA++ (Brown et al. 1990, 1993; Och 2000). Instead, we used the functionality provided by the hybrid alignment tool ATLAS: it has been designed specifically to use a variety of alignment strategies and to align even extremely small parallel corpora (Schrader 2006).

Table 2. Statistics on Word Alignment

Lang. Pair	Minimum	1st Qu.	Median	Mean	3rd Qu.	Max.	Std. Dev.
OMD-NURN	0.29	0.67	0.83	0.80	1.00	1.00	0.17
OMD-REG	0.20	0.66	0.78	0.78	1.00	1.00	0.18
OMD-AUG	0.40	0.63	0.73	0.75	0.88	1.00	0.17
OMD-SCHW	0.29	0.67	0.82	0.80	1.00	1.00	0.17
NURN-OMD	0.29	0.67	0.83	0.80	1.00	1.00	0.17
NURN-REG	0.27	0.77	0.90	0.86	1.00	1.00	0.17
NURN-AUG	0.36	0.70	0.85	0.82	1.00	1.00	0.16
NURN-SCHW	0.40	0.75	0.86	0.85	1.00	1.00	0.14
REG-OMD	0.20	0.66	0.78	0.78	1.00	1.00	0.18
REG-NURN	0.27	0.77	0.90	0.86	1.00	1.00	0.17
REG-AUG	0.25	0.71	0.83	0.81	1.00	1.00	0.17
REG-SCHW	0.30	0.71	0.83	0.81	1.00	1.00	0.16
AUG-OMD	0.40	0.63	0.73	0.75	0.88	1.00	0.17
AUG-NURN	0.36	0.70	0.85	0.82	1.00	1.00	0.16
AUG-REG	0.25	0.71	0.83	0.81	1.00	1.00	0.17
AUG-SCHW	0.43	0.67	0.82	0.80	0.94	1.00	0.15
SCHW-OMD	0.29	0.67	0.82	0.80	1.00	1.00	0.17
SCHW-NURN	0.40	0.75	0.86	0.85	1.00	1.00	0.14
SCHW-REG	0.30	0.71	0.83	0.81	1.00	1.00	0.16
SCHW-AUG	0.43	0.67	0.82	0.80	0.94	1.00	0.15

cases, the mean of the similarity scores, computed over all word alignments in a language pair, ranges between 0.75 and 0.86. Similarly, the standard deviation is relatively small in all cases. The median, however, shows a more ragged distribution with a minimum value of 0.73 and a maximum of 0.90, which we take as indication that there are systematic differences between the five dialects. Still, when taking the statistics at face value, all five dialects appear to be highly similar.

However, if we rank the language pairs using the mean, we can hypothesize that on average, the OMD text is most dissimilar to all other texts of our case study: in all pairwise comparisons, the pairing involving OMD receives a mean that is considerably lower than the means of all other language pairs. NURN represents the opposite of OMD by receiving the highest means in all pairings, i.e. it appears to be most similar to all other texts. We observe further that the three Bavarian texts NURN, REG, and AUG appear not to be overly similar.

An analysis of the standard deviations basically confirms these findings. We assume that a low standard deviation reflects a high similarity between languages based on the intuition that if two languages are, on the whole, very similar, then the variation around the mean similarity score should be low too. Thus, OMD is again clearly shown to be most dissimilar to all other languages of our sample, language pairs involving OMD always having the highest standard deviations. On the opposite extreme, NURN and SCHW have the lowest standard deviation of all language pairings, thus we may hypothesize that the vocabularies of these dialects, at least in our corpus, are most similar.

3.2 Qualitative analysis

Following the statistical analysis of the word alignment, we also inspected some of the data qualitatively, in order to (i) evaluate the quality and hence reliability of the automatic word alignment, and (ii) analyze which types of graphemic variation occurred, and hence assess whether our automatic procedure is suitable to give insights into how similar or different the involved dialects are.

With respect to the alignment quality, we exploited part-of-speech information (see Sec. 4) to extract possibly erroneous word pairs: if a word pair shows part-of-speech mismatches, as e.g. in example (1), we have reasons to assume that these category mismatches are either due to real category changes, and, hence, possible divergences between the two languages, or due to erroneous word alignment. In sum, we found out that 12.44% (254) of all word pairs (2042) showed category mismatches. Most of these are indeed due to alignment errors (185, or 9.06%). Others, however, point to minor tagging mismatches as in example (2), which may or may not reflect grammatical differences.

(1) *leut* (noun) – *waren* (auxiliary)

(2) *die* (pronoun) – *dí* (determiner)

Exemplarily, we also analyzed all those word pairs in the language pair OMD–NURN[7] that had a similarity of 0.8 or above (excluding exact matches, i.e. word pairs with a similarity of 1).

In this small data sample, we found 40 word pairs that are overall similar, but simultaneously show systematic variation:

(3) *zu/tzu, juda$z/judas, jo$zeph/io$eph, kinde$z/kindes*

(4) *wil/will, hatten/heten, fur$ten/für$ten, $i/$ie, gieng/ging, bruderen/prudern*

(5) *girig/geitig* (lexical), *sach/gesach* (morphological)

(6) *vnde*, *vnd* and *vnn*; *pfennige* and *phennige*

Probably, the examples in (3) can be interpreted as being due to different writing conventions. On the other hand, the examples in (4) most probably reflect real phonetic

[7] Assuming that the dissimilarities of this dialect pair, as statistically shown above, should allow us to gain valuable insights.

differences. We also observed word pairs that indicate lexical and morphological differences between the two dialects (5). Finally, we found instances of inconsistent spelling within the same text, e.g. the OMD text contains all spelling variations shown in (6). Such inconsistencies could indicate ongoing *changes* in a dialect's system, or changes that have occurred rather recently, resulting in spelling uncertainties.

Although the above analysis is rather sketchy, and restricted to very few word pairs, they are sufficient to highlight the usefulness of our methodology: without investing too much effort in a sophisticated procedure, we are able to automatically word-align small text samples of historical dialects. We are using a very simple similarity measure that relies exclusively on the graphemic similarity of the tokens in the parallel corpus. The error rate of the word alignment procedure seems to be quite low. Furthermore, the similarity measure also provides means for (i) comparing the dialects as a whole, in order to determine whether, on principle, two dialects are similar, and (ii) to extract remarkable word pairs from the corpus for a detailed linguistic analysis.

4 Similarity based on ngrams

In addition to computing similarities based on word-alignment measures, we ran another series of experiments based on frequencies of character ngrams (4.1) and part-of-speech tag ngrams (4.2). The ngram models serve us as *text fingerprints* or *profiles*, which are compared one to another.

4.1 Character ngrams

Ngram statistics are widely used in natural language processing. Most often, ngrams consist of sequences of words or part-of-speech tags, less frequently ngrams of *characters* are used. Character-based ngram models have been applied to a range of tasks that involve similarity computations, such as spelling error detection (Zamora et al. 1981), authorship attribution (Fuchun et al. 2003; Kjell 1994), language or topic classification (Beesley 1988; Cavnar and Trenkle 1994), and information retrieval in general (Damashek 1995).

In our study, we compared the distributions of character ngrams, with n=1,2,3, across the texts. As an example, Table 3 lists the top-ranked character bigrams across the texts, along with their absolute frequencies and their weights, which determine the ranking in the table. $W_{t,n}$, the weight of an ngram n in a text t, is computed as $W_{t,n} = \frac{F_{t,n}}{F_{T,n}}$, where $F_{t,n}$ is the frequency of an ngram in a specific text, and $F_{T,n}$ is the frequency of the ngram in the entire collection (i.e., in all five texts). $W_{t,n} = 1$ means that the ngram only occurs in the current text and, hence, can be interpreted as a

Table 3. Top-ranked character bigrams across the five texts, with absolute frequencies and weights

OMD			NURN			REG			AUG			SCHW		
a$	10	1.00	iu	5	0.62	sy	7	1.00	lí	6	1.00	vd	7	1.00
$z	26	1.00	pf	8	0.57	aw	5	0.62	eí	16	1.00	iv	9	1.00
th	10	0.67	as	11	0.50	ai	5	0.62	zz	5	1.00	$v	8	1.00
tt	8	0.67	fe	8	0.50	un	10	0.62	dí	12	1.00	d@	11	1.00
eb	10	0.53	$$	8	0.44	ff	5	0.42	aí	5	1.00	vi	18	0.82

characteristic feature of this text (and, ideally, this can be related to features specific to the dialect used in this text). That is, with regard to writing conventions, Text AUG and SCHW stand out due to their idiosyncratic spellings. In contrast, Text NURN exhibits a sort of "average spelling" (similar observations have been made in Sec. 3).[8]

4.2 POS unigrams and bigrams

Character ngrams can give hints as to whether two texts share a certain amount of vocabulary and (graphemic-)phonetic and inflectional features. They certainly cannot be used for syntactic comparisons. Lüdeling (2006) used syntax trees in her study; in contrast, we use POS ngrams as a cheap surrogate for syntax. We manually annotated all texts with POS tags according to the STTS tagset.[9]

[8] The tendencies that one can observe with bigrams also show up with the other ngram types. This is partly related to the way we compute the scores: one of the reasons of, e.g., *d@* being a sequence unique to Text SCHW is the fact that we introduced the letter @ only in this text, see Fn. 5. Similarly, the character *í* only occurs in Text AUG, but with high frequency. A more sophisticated ngram weighting measure would therefore take the frequencies of (n-1)grams into account.

[9] `http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-1999.ps.gz`.
The STTS tagset has been developed for the annotation of modern German. In applying the tagset to our texts, we encountered surprisingly few problematic cases: Distinguishing adverbs (ADV) from verb particles (PTKVZ) can be difficult, as in *vnde $ie von dem ti$che* ***vf*** *waren ge$tanden* ('and they had stood up from the table', OMD). Pronominal adverbs such as *davon* 'thereof' are mostly spelt in two words, which we annotated as *do/ADV von/APPO*. For seemingly noun compound constructions, we defined a new tag PTKNZ ("Partikel Nomenzusatz", 'noun particle') for the non-head components: *erb/PTKNZ $chafft/NN* 'heir_ship', *nach/PTKNZ kommen/NN* 'off_spring', *an/PTKNZ vanch/NN* 'begin'.
A final example is the distinction between demonstrative and relative pronouns, which in modern German is done on the basis of the verb position (verb second vs. verb final). However, the evolution of the modern verb position is a research question in itself, so no prior decision should be made during POS annotation. A relevant example from our texts is *do kaufften in einer hand leut* ***die*** *hie$$en y$mahelite* ('Then some people bought him, who were called Ysmaelite/They were called Ysmaelite.', NURN). We currently use underspecified tags for these cases: *die/PDS_PRELS*.

Table 4. Top-ranked POS unigrams (top) and bigrams (bottom) across the texts, along with their absolute and relative (%) frequencies

POS Tag	OMD		NURN		REG		AUG		SCHW	
NN	42	18.03	44	17.96	42	18.10	43	19.20	45	18.91
PPER	22	9.44	26	10.61	26	11.21	21	9.38	25	10.50
ART	23	9.87	19	7.76	20	8.62	20	8.93	20	8.40
ADV	16	6.87	20	8.16	15	6.47	16	7.14	21	8.82
VVFIN	13	5.58	18	7.35	17	7.33	16	7.14	17	7.14
VAFIN	16	6.87	16	6.53	14	6.03	14	6.25	15	6.30
Sum	233	100	245	100	232	100	224	100	238	100
ART+NN	21	8.97	15	6.10	17	7.30	17	7.56	17	7.11
PPOSAT+NN	3	1.28	10	4.07	7	3.00	9	4.00	9	3.77
ADJA+NN	8	3.42	2	0.81	6	2.58	4	1.78	4	1.67
Sum	234	100	246	100	233	100	225	100	239	100

One way of exploiting the POS information would be to align these tags across the texts and compute similarities just as described in Sec 3. Another way, however, is to explore properties of the POS annotations themselves and compare *patterns* of POS annotations across the texts. In a similar vein, Lauttamus et al. (To Appear) applied this method to the classification of language data produced by first vs. second language learner.

POS ngrams distribute much more evenly across the texts than character ngrams, so we can present an overview of the rankings across all texts. Table 4 lists the top-ranked POS unigrams and bigrams, along with their frequencies.

As can be seen from Table 4, all texts except OMD show similar rankings with regard to unigrams and bigrams: ranks 1-2 are occupied by NN and PPER, respectively. Ranks 3–5 are taken by ART, ADV, and VVFIN, in slightly varying orders. With bigrams, Texts 2–5 show identical rankings. In contrast, OMD ranks ART second rather than PPER, and prefers ADJA+NN over PPOSAT+NN among the bigrams (these tag sequences occur e.g. in the formulae *mein/PPOSAT kint/NN* (‘my child’) and *mein lieber/ADJA Son/NN* (‘my dear son’).

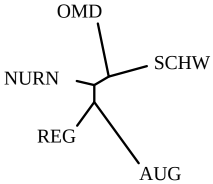
4.3 Computing similarity and relatedness

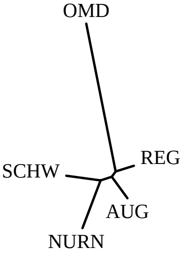
The character and POS ngrams are used to compute pairwise similarity between the texts. For computing similarity, we applied the cosine measure. Pairwise similarity measures result in a similarity matrix, cf. Table 5. Based on the similarity matrix, we computed a phylogenetic tree, using the *Neighbor Joining Method* (Saitou and Nei

Table 5. Similarity matrices for pairwise similarities based on character (top) and POS (bottom) bigrams and the corresponding phylogenetic (unrooted) trees

Char. Bigrams	OMD	NURN	REG	AUG	SCHW
OMD	100	88.76	85.93	80.79	88.61
NUERN		100	93.44	85.31	91.40
REG			100	87.27	86.12
AUG				100	82.47
SCHW					100

POS Bigrams	OMD	NURN	REG	AUG	SCHW
OMD	100	80.73	85.79	84.69	83.67
NUERN		100	92.80	93.18	92.84
REG			100	95.56	93.94
AUG				100	93.25
SCHW					100





1987). This algorithm first relates the least distant pair, merges them, and applies the algorithm to the remaining pairs plus the newly created merged one. The result can be visualized, e.g., by unrooted trees. Table 5 shows the similarity matrices along with the corresponding phylogenetic trees that cluster character and POS bigrams.[10]

The results shown in Table 5 confirm our observations made with regard to the rankings of the top-ranked ngrams in Tables 3 and 4: (i) At the level of character bigrams, AUG and OMD are the most idiosyncratic texts, with values ranging between 80.79–87.27 (AUG) and 80.79–88.76 (OMD). Hence, AUG and OMD are located on rather long isolated branches of the phylogenetic tree, while NURN is the most "average" one (with values of 85.31–93.44), and therefore located in the center of the tree. (ii) At the level of POS bigrams, OMD clearly stands out, while the other texts cluster nicely. With neither of the measures would the three Bavarian texts, NURN, REG, AUG, form a cluster on their own, again as already observed in the alignment experiment, Sec. 3.

[10] The length of the branches indicates the degree of distance. The scale used in both figures are different, though. The trees have been created by the software package PHYLIP (Felsenstein 1989).

5 Conclusion

The goal of this paper was to investigate whether quantitative methods can be sensibly applied to small text samples of historic German dialects. We showed first that a simple, automatic word alignment procedure can be used to align the data successfully. We then used the word alignment and character and POS ngram models to detect similarities and differences between these writing dialects, and to compute clusters of dialects. One of the dialects, OMD, was clearly shown to be most dissimilar to the other four dialects. This result correctly reproduces the well-known distinction between Middle and Upper German.

The four Upper German dialects, on the whole, could be shown to be highly similar by the similarity computations based on word-alignment and POS ngrams. In contrast, character ngrams singled out AUG as highly idiosyncratic, closely followed by OMD. The three Bavarian texts could not be shown to form a cluster, which might be attributed to the fact that only REG represents a "canonical" Bavarian text.

Finally, with the string-based methods (word-alignment and character ngrams) NURN came out as the most "neutral" dialect, sharing many characters and character sequences with the other dialects.

Our next steps, in the context of a larger project, will be to expand our data to include up to 50 complete corpus samples. We also plan to repeat our word alignment analysis using a more refined similarity measure which incorporates linguistic knowledge on sound changes ("Lautgesetze"). Similarly, in a sort of bootstrapping approach, we would integrate prior findings (like the correspondence of *\$\$–zz* in certain text pairs) into the alignment procedure and use this information in subsequent alignments of the respective texts.

Finally, it is planned to more thoroughly investigate into the historical origins of the data, especially with respect to authorship and translational routes from one dialectal text to the next. Thus, we hope to determine which similarities of the data are possibly due to "translationese" that occurred when one author copied, or even translated, the texts from one dialect into the other. Ultimately we would like to be able to automatically determine which of the similarities between two texts are artifacts of the copy procedure, and which ones derive from genuine linguistic proximity of the dialects involved.

References

Beesley, Kenneth R. (1988). Language Identifier: A Computer Program for Automatic Natural-Language Identification of on-line Text. In *Languages as Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association*, 47–54.

Besch, Werner (1967). *Sprachlandschaften und Sprachausgleich im 15. Jahrhundert. Studien zur Erforschung der spätmittelhochdeutschen Schreibdialekte und zur Entstehung der neuhochdeutschen Schriftsprache*. München.

Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin (1990). A Statistical Approach to Machine Translation. *Computational Lingusitics* 16(2):79–85.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer (1993). The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics* 19(2):263–311.

Cavnar, William B. and John M. Trenkle (1994). N-Gram-Based Text Categorization. In *Proceedings SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval.*

Damashek, Marc (1995). Gauging Similarity with n-Grams: Language-Independent Categorization of Text. *Science* .

Felsenstein, Joseph (1989). PHYLIP — Phylogeny Inference Package (Version 3.2). *Cladistics* 5:164–166.

Fuchun, Peng, Dale Schuurmans, Vlado Keselj, and Shaojun Wang (2003). Language Independent Authorship Attribution Using Character Level Language Models. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, Budapest, Hungary.

Gray, Russell D. and Quentin D. Atkinson (2003). Language-Tree Divergence Times Support the Anatolian Theory of Indo-European Origin. *Nature* 426:435–439.

Kjell, Bradley (1994). Authorship Determination Using Letter Pair Frequency Features with Neural Network Classifiers. *Literary and Linguistic Computing* 9(2):119–124.

Lauttamus, Timo, John Nerbonne, and Wybo Wiersma (To Appear). Detecting Syntactic Substratum Effects Automatically in Interlanguage Corpora. In Muriel Norde, Bob de Jonge, and Cornelius Hasselblatt (eds.), *Language Contact in Times of Globalization*, Amsterdam: Benjamins.

Lüdeling, Anke (2006). Using Corpora in the Classification of Language Relationships. *Zeitschrift für Anglistik und Amerikanistik. Special Issue on 'The Scope and Limits of Corpus Linguistics'* 217–227.

Mosteller, Fred and David Wallace (1964). *Inference and Disputed Authorship: The Federalist Papers.* Massachusetts:Addison-Wesley.

Nerbonne, John, Wilbert Heeringa, Eric van den Hout, Peter van de Kooi, Simone Otten, and Willem van de Vis (1996). Phonetic Distance between Dutch Dialects. In *Proceedings of the Sixth CLIN Meeting*, 185–202, Antwerp.

Och, Franz Josef (2000). Giza++: Training of Statistical Translation Models. `http://www-i6.informatik.rwth-aachen.de/~och/software/GIZA++.html`.

Saitou, Naruya and Masatoshi Nei (1987). The Neighbor-Joining Method: a New Method for Reconstructing Phylogenetic Trees. *Molecular Biology and Evolution* 4:406–425.

Schrader, Bettina (2006). ATLAS – A New Text Alignment Architecture. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 715–722, Sydney, Australia: Association for Computational Linguistics.

Spruit, Marco Rene (2006). Discovery of Association Rules between Syntactic Variables — Data Mining the Syntactic Atlas of the Dutch Dialects. In *Computational Linguistics in the Netherlands.*

Swadesh, Morris (1955). Towards Greater Accuracy in Lexicostatistic Dating. *International Journal of American Linguistics* 21:121–137.

Zamora, E. M., Joseph J. Pollock, and Antonio Zamora (1981). The Use of Trigram Analysis for Spelling Error Detection. *Information Processing & Management* 17(6):305–316.

Data structures for the analysis of regional language variation

Birgit Kellner, Timm Lehmberg, Ingrid Schröder and Kai Wörner

Abstract. This article reports on work in progress on the development of data structures and processing methods which take into account the special demands for the documentation and analysis of regional language variation. This work is an integral part of the supra–regional project "Language Variation in Northern Germany" which started in the beginning of 2008 as a joint initiative of six Northern German universities.

1 Introduction

The project "Language Variation in Northern Germany" ("Sprachvariation in Norddeutschland" — SiN) aims at the documentation, description and analysis of the usage of dialectal variation in Northern Germany along a spectrum between Low German and High German.

This objective requires the analysis of object language material from communication settings with different degrees of formality as well as metalinguistic information on speaker biographies, language awareness and language attitudes. The data used for this purpose is gathered through various settings of acquisition (translation, interview, spontaneous talk, salience tests etc.) and thus is highly heterogeneous and complex with regard to the respective level of dialectological analysis. It is the basis of a deeply annotated over–all Northern German corpus of spoken language. The implementation of such a corpus forces the use of sophisticated data standards and tools for the processing (recording, transcriptions, annotation etc.) and analysis of spoken language.

The article is structured as follows: Section 1 gives an overview of the research problem as well as the chronological and geographical structure of the project. Based on this, the methods and principles of data acquisition are presented. Section 2 shows the present state of the implementation with assistance of the EXMARaLDA system which is used for the processing of the entire language data. A special focus is laid upon the modelling of the metadata and the problems of a multi–layer annotation of transcribed spoken language.

2 The project

2.1 Starting point and aim

Within the project “Language Variation in Northern Germany”, for the first time the entire spectrum of (autochthonous) languages in the Northern German dialect area is described and analyzed with regard to the different regional varieties. This happens on the basis of a spoken corpus which is being gathered according to standardized criteria of data acquisition. In doing so the project’s conception conforms to the principles of *new dialectology* (cf. Elmentaler et al. 2006; Niebaum and Macha 2006). In contrast to traditional dialect research, which aims at a retrieval of the most basic forms of dialectal speech, this discipline focuses on the description of the entire spectrum of variations between the two poles standard language and basic dialect (Schröder 2004: 82). At the same time the project aligns with a new type of language geography that has been developed, for instance, the multi–dimensional language atlases by Bellmann et al. (1994–2002) (cf. Elmentaler 2006).

As a speciality of the project, for the first time in dialect research, a supra–regional comparative analysis of spontaneous speech in a natural context of interaction (language *in vivo*) is introduced. Thereby the different levels of the subjects’ language strata are elicited with reference to communication situations of different degrees of formality — ranging from highly formal to informal. It is of interest, if there are either distinct language varieties between dialect and standard language, or there is a continuum of merging language strata without a clear distinction between these two poles. (The latter case would have impact on the status of Low German as an autonomous language.) Furthermore especially the conditions and regulations of language choice and language use, such as code–switching, code–mixing and punctual interference, have to be described. To give information on the individual language profiles of the subjects, their language socialization, language attitudes and language awareness will be included into the survey.

The fact that the project covers the entire region of Northern Germany for the first time enables the examination of individual and regional specific differentiation of language strata as well as the perception and classification of salient features of everyday language. In the context of areal language change, this makes it possible to elucidate the mutual relations between structural, pragmatic and psycho–social factors.

Apart from the above mentioned structural description of language strata on the basis of *object language data*, the regional as well as pragmatic variables that have influence on Northern German everyday language have to be determined. In doing so, the focus is laid on *metalinguistic data* that is elicited through various acquisition settings[1] (see also section 1.3). These information on, amongst others, a subject's language biography as well as language awareness, language perception and language attitudes not only form an important basis for the documentation of the status quo of the dialect situation but also for a prognostication of the further development of spoken Low German in the different regions of Northern Germany (cf. Pochmann 2007: 23). Thus, the project meets the demands of a complementary explanation of structural data by subjective speaker data.

2.2 Research phases

Due to the high degree of dialectal variety in Northern Germany and the enormous effort in data acquisition, the project has to be realized through a joint initiative of research institutes from six Northern German universities (see below), each dealing with different aspects of dialectological research. Chronologically, the project is structured into two research phases. During the first phase (2007–2009), the data is being acquired and processed according to standardized conditions at all six research locations. In the second phase (2009–2011), the annotation and analysis of this data is carried out by all six sites with regard to its respective main research.

Phase 1: data acquisition

The information is gathered through six acquisition settings (see section 1.3) consisting of interviews, several tests of language awareness and language attitudes and the analysis of recorded informal conversation. As part of each setting, the entire process of data acquisition is documented by audio recordings that later are transcribed and annotated according to several issues of dialectology.

For this purpose, 36 acquisition locations spread over 18 regions, each with distinct characteristics of dialect structure have been chosen. Each of the six research sites collects data in three regions at two selected villages. They have been chosen with regard to their demographic and economic profile; they count between 2000 and 8000 inhabitants and have a rural structure. At each of the 36 locations, data will be gathered from four female subjects at the age between 40 and 55 who have been born, raised and lived for the most part of their life at the acquisition location. If possible, at each acquisition location, data will be gathered from two subjects with good dialect competence and two subjects without any dialect competence.

[1] The term *metalinguistic data*, as it is used here, refers to a subject's perception and evaluation of the own and others' language use and language acquisition. Though there are certain correspondences to speaker related *metadata*, the two terms have to be kept distinct.

Phase 2: data annotation and analysis

During the second phase, the object language data and metalinguistic data gathered in phase 1 is annotated and analysed under three different aspects by sub–projects put together from the above mentioned six research sites.

- **Sub–Project 1** (University of Kiel, University of Frankfurt(Oder)) analyses the language strata of the lower (dialect based) and the upper (standard convergent) spectrum under quantitative areal linguistic aspects. This will be done with regard to the different regional characterisations of the elicited Northern German language varieties as well as inter–regional tendencies of assimilation within the dialectal language strata.

- **Sub–Project 2** (University of Hamburg, University of Münster) introduces a qualitative analysis of the individual speaker related characteristics of language strata profiles (degree of dialecticity) in a supra–regional comparison of the object language data Under aspects of pragmatics, the mixing of language strata such as code–switching, code shifting and code fluctuation are analysed with consideration of the functional distribution and the situational contexts of their usage.

- **Sub–Project 3** (University of Bielefeld, University of Potsdam) analyses the metalinguistic data under aspects of sociolinguistics. The research concentrates on language acquisition, conditions of language use and language shift, respective regularities of the use of different language varieties, and finally, language awareness and language attitudes. The focus lies on an analysis of the interrelation between dialectal phenomena and the subjects' regional origin.

2.3 Data acquisition settings

To achieve an optimal documentation of the different language strata along the spectrum of standard language, dialect and convergent varieties, the data is collected through the following six acquisition settings:

1. To identify their maximum dialectal language stratum, subjects are requested to recite 40 pre–defined phrases which have been established in dialectology (the so–called *Wenker Phrases*), by using their respective regional basic dialect.

2. *Spontaneous talk* using the dialect enables to assess the subject's dialectal competence in an informal situation.

3. An *interview* conducted in High German allows for an elicitation of both a medium language stratum on the object language level and sociolinguistic data on the metalinguistic level.
4. The *reading aloud* of a High German text by the subjects, who are asked to perform as standard convergent as possible, will enable researchers to elicit their maximum standard convergent language stratum.
5. The recording of a *family conversation* allows for an examination of the use of spontaneous speech in interaction.
6. In addition to these five settings and with regard to the issues of sub–project 3 (see section 1.2), four different tests for the analysis of a subject's individual language awareness, language attitudes and linguistic evaluation have been put together in one setting:
   (a) A *salience test* to examine the subject's perception of local or regional deviation from the standard language.
   (b) A *situational judgement test* to ask for the appropriateness of recognized salient features in situations with different degrees of formality.
   (c) A *normativity test* to examine the subject's own acceptance of the recognized salient features in conversations with a higher degree of formality.
   (d) Tests on a subject's *mental maps*.

3 Data modelling

As an essential part of each data acquisition setting described in section 1.3, audio data is recorded, transcribed and annotated with respect to the above mentioned issues of dialectological research. In conjunction with the elicited metalinguistic data (see section 1.1) and comprehensive metadata on subjects, recordings, transcriptions as well as communication contexts, they constitute a cluster of information that puts together different layers of dialectical analysis. The primary task of an underlying data model is to link these layers to each other and to allow for a global analysis of the data. For this purpose audio data needs to be synchronized with transcriptions and several annotation layers (phonetic, morphologic, syntactic discourse annotation etc.). The annotation layers in turn must be correlated with the above mentioned metadata on speakers and conversation situations, but also with information on the subject's perception of normativity and salience.

Furthermore the data format of the corpus must allow for a sustainable storing and accessing of the collected data. Therefore it ideally has to be compatible and interoperable to existing and widely adopted standards for the preparation of linguistic resources (cf. Rehm et al. 2008; Schmidt et al. 2006).

At the present point of time, EXMARaLDA is the only system that on the one hand fits the demands of sustainability and interoperability and on the other hand provides a number of tools for structuring, annotating and visualizing spoken language data, multi–layer annotation and specific metadata for spoken language corpora. In the following section, the core components of EXMARaLDA will be introduced with a focus in the EXMARaLDA metadata schema. Subsequently an overview of the current state of the implementation of the data model is given.

3.1 The EXMARaLDA system

EXMARaLDA ("Extensible Markup Language for Discourse Annotation") is a system of concepts, XML–based data formats and tools for the computer assisted transcription and annotation of spoken language and for the construction and analysis of spoken language corpora (cf. Schmidt 2005, 2007; Schmidt and Wörner 2005). It is being developed at the Collaborative Research Center "Multilingualism" (Sonderforschungsbereich "Mehrsprachigkeit" — SFB 538) at the University of Hamburg. All components of the EXMARaLDA system are freely available to all users and run on all major operating systems.

The core applications of the EXMARaLDA System are:

- The *Partitur Editor*, a tool for the input, editing and output of transcriptions in a musical score (*Ger.*: "Partitur") notation. It has import and export features for the data exchange with other transcription systems like TASX, Praat, ELAN, AIF or syncWRITER. Through the underlying, graph–based XML–format, it offers numerous export, visualisation and processing options.

- The *Corpus–Manager* (CoMa), an application that enables researchers to link transcriptions created with the EXMARaLDA Partitur Editor with their corresponding recordings into corpora and to enrich them with comprehensive metadata.

- The *EXMARaLDA Analysis and Concordancing Tool* (EXAKT), a tool for the searching and retrieval of transcribed and annotated phenomena from an EXMARaLDA corpus. The identified results can be contextualized and processed in multiple ways.

The underlying schema for CoMa Metadata is based on communication events (referred to as "communications") and persons (referred to as "speakers"). Communications are assigned to audio or video recordings and their transcriptions, speakers are linked to the communications in which they play an active role. In doing so, metadata for speakers that appear in multiple communications has to be entered just once. In contrast to a setting where all metadata is stored in the transcriptions headers, this approach enables researchers to avoid errors and redundancies and provides a better basis for querying and filtering the data.

For constellations that already provide rich metadata in the EXMARaLDA–transcription–headers, CoMa can use these data to generate corpora with the corresponding metadata through an automated import routine. However, the metadata in the EXMARaLDA–transcriptions has to be in a condition that assures that transcriptions can unambiguously be mapped to communications and each speaker can be identified distinctly.

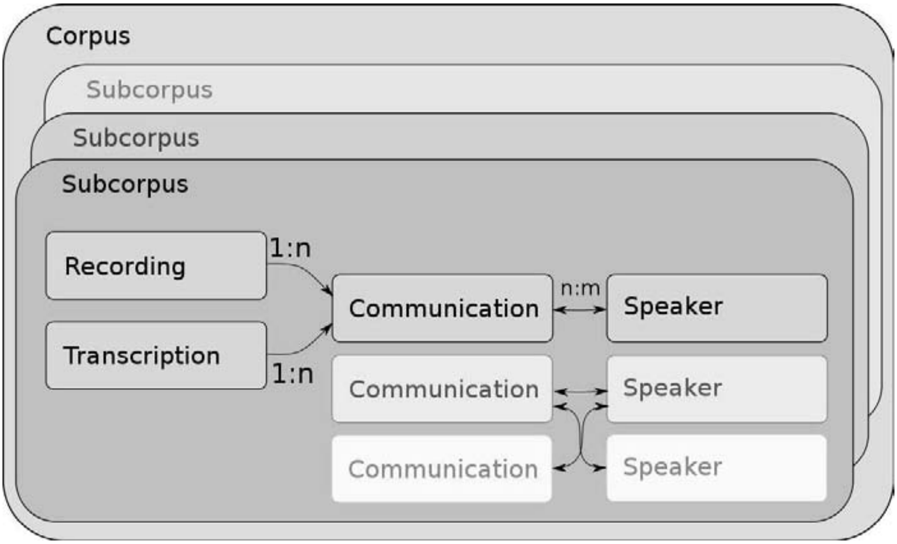


Figure 1. The CoMa Metadata Schema

The CoMa metadata schema determines only a few metadata elements (like sex and speaker abbreviation). An unlimited number of metadata entries can be entered through free attribute–value–pairs whereby the attribute–vocabulary should be fixed for all speakers and communications of the same corpus or subcorpus. This openness was chosen deliberately, as practical experience showed that fixed metadata schemas of any extent often do not match the corpus design, either because they are too potent or because they lack important features (for an overview of existing standards for the storage of metadata, see Lehmberg and Wörner (2008)). Users are free to use existing metadata vocabularies, though, since they can easily be mimicked through CoMa's free attribute–value pairs.

Metadata gathered in CoMa can be filtered in multiple ways to gain information about the composition of the corpus, the speakers involved and many things more. The filter mechanism can be used to generate subcorpora based on metadata categories, which can be searched by the query–tool EXAKT.

3.2 The SiN metadata schema

A metadata schema that fits the demands described in section 1 requires consideration of interdependencies between subjects and their social environment, both in the recorded conversation and over the entire period of language socialization. Beyond this there is a large number of metalinguistic information on the perception of subjects concerning their personal usage of dialect as well as their dialect acquisition. This specific information in some cases is being elicited directly from the primary data in the framework of acquisition scenarios like family conversation and spontaneous talk. (It is, for instance, easily conceivable that subjects report on the stigmatization of dialect usage at school or an intended usage of their respective dialect in the framework of family celebrations or local clubs and societies).

Due to the individual characteristics that result from the above mentioned interdependencies, an approach on modelling all information on a subject by means of one single metadata record would lead to a large number of miscellaneous and individual metadata entries. To keep the metadata schema as manageable and transparent as possible, based on the CoMa Metadata model, all persons with influence on a subjects dialect acquisition are registered as speakers, regardless if they participate in the recorded conversations (especially family conversation) as active speakers or if they are only mentioned by a subject (for instance, in the framework of the interview). Links to the respective subject are generated by a relational entry. In doing so, individual constellations of interdependencies between speakers can be modelled (and later analysed) in a straightforward way, using only a small amount of metadata fields.

By predefining location–elements like birth, education, residence, life partnership etc., all relevant stages of a speaker's life that might have impact on their dialect acquisition can be stored in the metadata. Using the CoMa metadata model, speakers can be linked to Communication–entities that are used to represent the six acquisition settings (interview, family conversation etc., see Fig. 2).

Concerning the storage of the elicited metalinguistic information (cf. section 1.1) by means of metadata entries, the situation becomes much more sophisticated. As mentioned above, the CoMa data model only provides the storage of information on speakers, communications, recordings and transcriptions. Metalinguistic information, however, appears within the frame of sentences and utterances. For this reason, the only possibility to store this information is to predefine an annotation vocabulary (for instance, *school*, *dialect stigmatization* etc.) and to add this information to the

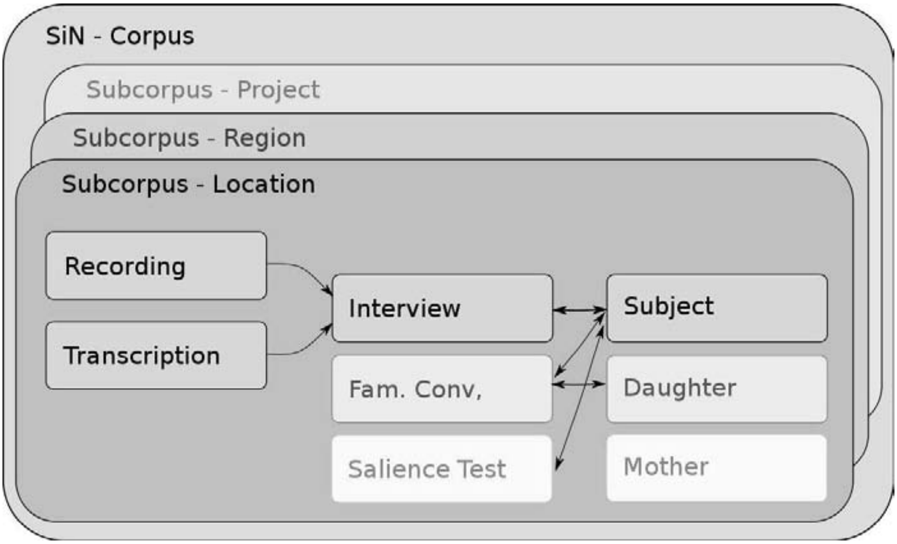


Figure 2. Implementation of the CoMa Metadata Schema

respective segments by using additional annotation tiers.

3.3 The SiN annotation schema

As described above, the intended analysis aims at considering the entire spectrum of dialectal characteristic that is phonological, morphological but also syntactic and pragmatic phenomena. Depending on the respective acquisition setting, these phenomena as well as information on a subject's perception of salience, code–switching and the abovementioned content–related analysis have to be stored as different annotation layers on annotation tiers, using the EXMARaLDA Partitur Editor. The fact that the annotation procedure on the one hand has to focus on various levels of dialectal analysis and on the other hand takes into account all common levels of language description, leads to several problems in defining unique annotation layers. Whereas in the frame of a structured multi–layer annotation usually each annotation layer corresponds to one level of language (i.e. morphology, syntax, pragmatics etc.) that can be annotated and analyzed separately, an annotation of dialectal phenomena, as described here, can lead to an overlap of these layers. As an example, in some cases there might be an indifferent mapping of dialectal phenomena to a morphological or phonological annotation layer.

There is also the fact that a separation of the annotation of dialectal phenomena into annotation layers are based on levels of language, would lead to a large and unmanageable number of annotation tiers, each containing only little information. However, merging all dialect–specific information to one single annotation layer would cause an overlap of the annotated segments of speech (tokens, phrases, utterances and sentences).

Against this background the following two approaches to this problem are discussed:

1. Breakdown by level of dialectal analysis:

   Layer 1: structural annotation, dialectal characteristics and dialecticity
   (segmentation on token–level)
   Layer 2: pragmatic annotation, annotation of code–switching etc.
   (segmentation on utterance–level)
   Layer 3: content analysis, metalinguistic information etc.
   (segmentation across utterance–level)

2. Breakdown by segment size:

   Layer 1: annotation of phonological and morphological phenomena
   (segmentation on token–level)
   Layer 2: syntactical and pragmatic annotation
   (segmentation across token–level)
   Layer 3: content analysis, metalinguistic information, code–switching etc.
   (segmentation across utterance–level)

Another challenge is posed by the following step that is defining a tagset for a cross–regional and in the strict sense multilingual corpus. Instead of using individual annotation vocabularies for every single region/dialect, the project aims at creating an over–all Northern German tagset containing a complete vocabulary for all expected dialectal phenomena.

4 Conclusion

The approaches to the problems of planning and implementing a highly heterogeneous and deeply annotated corpus of dialect language as touched upon in this article show a number of characteristics that are typical for the annotation and analysis for spoken corpora in general. These are, amongst others, the possibility of eliciting comprehensive metadata as well as metalinguistic information from primary data, intermixture of metadata and annotation and general problems in mapping the levels of analysis to annotation layers.

Due to the specific infrastructure, the comprehensive preliminary work as well as the data standards that are used from the very beginning, the project "Language Variation in Northern Germany" provides some good opportunities for new approaches to these issues.

References

Bellmann, Günter, Joachim Herrgen, and Jürgen Erich Schmidt (1994–2002). *Mittelrheinischer Sprachatlas (MRhSA).* Tübingen: Niemeyer.

Elmentaler, Michael (2006). Sprachlagenspektren im arealen Vergleich: Vorüberlegungen zu einem Atlas der deutschen Alltagssprache. *Zeitschrift für Dialektologie und Linguistik* 73:1–29.

Elmentaler, Michael, Joachim Gessinger, Jürgen Macha, Peter Rosenberg, Ingrid Schröder, and Jan Wirrer (2006). Sprachvariation in Norddeutschland. *Osnabrücker Beiträge zur Sprachtheorie* 71:159–178.

Lehmberg, Timm and Kai Wörner (2008). *Corpus Linguistics: An International Handbook*, chapter Annotation Standards. Berlin/New York: de Gruyter.

Niebaum, Hermann and Jürgen Macha (2006). *Einführung in die Dialektologie des Deutschen.* Narr, 2nd edition.

Pochmann, Christine (2007). *Sprachbiographien zwischen Hochdeutsch und Niederdeutsch in Finkenwerder. Wissenschaftliche Hausarbeit zur Erlangung des akademischen Grades eines Magister Artium der Universität Hamburg*. Master's thesis, University of Hamburg.

Rehm, Georg, Andreas Witt, Erhard Hinrichs, and Marga Reis (2008). Sustainability of Annotated Resources in Linguistics. In *Proceedings of Digital Humanities 2008, June 25–29, Oulu, Finland.*

Schmidt, Thomas (2005). *Computergestützte Transkription - Modellierung und Visualisierung gesprochener Sprache mit texttechnologischen Mitteln*, volume 7 of *Sprache, Sprechen und Computerand-Computer Studies in Language and Speech*. Frankfurt a. M.: Peter Lang.

Schmidt, Thomas (2007). Grundzüge von EXMARaLDA – einem System zur computergestützten Erstellung und Auswertung von Korpora gesprochener Sprache. In Jochen Rehbein and Shinichi Kameyama (eds.), *Bausteine diskursanalytischen Wissens*, Berlin: de Gruyter.

Schmidt, Thomas, Christian Chiarcos, Timm Lehmberg, Georg Rehm, Andreas Witt, and Erhard Hinrichs (2006). Avoiding Data Graveyards: From Heterogeneous Data Collected in Multiple Research Projects to Sustainable Linguistic Resources. In *Proceedings of the E-MELD 2006 Workshop on Digital Language Documentation: Tools and Standards: The State of the Art*, Lansing, Michigan.

Schmidt, Thomas and Kai Wörner (2005). Erstellen und Analysieren von Gesprächskorpora mit EXMARaLDA. *Gesprächsforschung* 6:171–195.

Schröder, Ingrid (2004). *Niederdeutsche Sprache und Literatur der Gegenwart*, chapter Niederdeutsch in der Gegenwart: Sprachgebiet — Grammatisches — Binnendifferenzierung, 35–97. Hildesheim.

Local syntactic tagging of large corpora using weighted finite state transducers

Jörg Didakowski

Abstract. This paper presents a new approach to fully automatic syntactic annotation of large corpora. The property of chunking to build local structures is adapted to syntactic tagging. By this a high degree of robustness is achieved. Here local dependency structures are represented by syntactic tags and bracketing and are built by a cascade of weighted finite state transducers. Competing structures are comparable by means of linguistic criteria formalized by a semiring. By this comparability of dependency structures preferences of syntactic readings and degrees of grammaticalness can be modeled. The scope of the local structures is defined by a longest match strategy. The approach is implemented and tested by means of the grammar driven parser SynCoP (syntactic constraint parser).

1 Introduction

Electronic text corpora are a basic tool for further development and evaluation in many natural language processing tasks. Furthermore, they are essential for the field of lexicography (cf. Sinclair 1991). Here the linguistic annotation plays a decisive role in exploring corpora and creating linguistic resources. In some applications linguistic annotated corpora of enormous size are needed. For example, in the creation of syntactic collocation databases large corpora are essential to achieve viable results (cf. Geyken et al. 2008; Kilgarriff et al. 2004). An annotation by hand or a semi-manual annotation is not practicable anymore. Instead, methods allowing a fully automatic annotation must be used. These methods must be robust and they have to cover the required linguistic depth at the same time. Furthermore, they must allow a very fast annotation process. Here it is assumed that the noise resulting from annotation errors can be softened by statistical approaches.

In this paper an approach is proposed which implements local syntactic tagging by means of constraint grammar tackling the problems of fully automatic annotation. The property of chunking (Abney 1990) of building local structures is adapted to syntactic tagging to achieve a high degree of robustness. The approach is based on Didakowski (2007), where the chunking is combined with the syntactic tagging treating chunks as local dependency structures.

Syntactic tagging by constraint grammars is successfully adopted in the annotation of large corpora as a reductionist approach (cf. Karlsson et al. 1995). In this approach the reductionist parsing can be performed by *finite state intersection grammars* (Koskenniemi 1990). But this reductionist approach has some disadvantages (cf. Tapanainen 1997, 1999; Voutilainen 1997): (1) The parsing works sentence-wise postulating sentence segmentation. (2) Preferences within global ambiguities and degrees of grammaticalness can not be modeled. (3) Rule writing is a balancing act: a rule should not forbid valid constructions and a rule should reduce ambiguities at the same time. The rules have to be written carefully which causes many unresolved ambiguities. (4) In the case of uncovered complex sentences, which can not be covered by an accordant finite state intersection grammar, an analysis can not be found. (5) In spite of a complexity of $O(n)$ intermediate results can grow rapidly during parsing causing computational problems.

In this paper an approach is presented tackling the above-mentioned problems. Syntactic tagging is implemented as a local method by sequentially applied weighted finite state transducers (WFST). First local scored dependency structures are generated on an input text by means of syntactic tags, bracketing and weights. The scores represent special characteristics of the corresponding structures. On the basis of the scores ambiguities can be resolved by means of linguistic criteria formalized by a semiring over the weights of the corresponding WFST.

The paper is organized as follows: In section 2 the basic definitions and notations are given. In section 3 the representation of the input and the analyses is defined. Hereupon the basic method is presented in section 4 which is extended by linguistic criteria in section 5, by a method for avoiding redundancies in section 6 and by a method detecting the maximum depth of dependency structures in section 7. Finally, the approach is implemented and tested with the analysis system SynCoP in section 8.

2 Definitions and notations

In the approach presented in this paper local dependency structures are generated and scored over an input by a WFST such that they can be judged by linguistic criteria. A weighted finite state transducer $T = (\Sigma, \Delta, Q, q_0, F, E, \lambda, \rho)$ over a semiring S is an 8-tuple such that Σ is the finite input alphabet, Δ is the finite output alphabet, Q is the finite set of states, $q_0 \in Q$ is the start state, $F \subseteq Q$ is the set of final states, $E \subseteq Q \times (\Sigma \cup \varepsilon) \times (\Delta \cup \varepsilon) \times S \times Q$ is the set of transitions, λ is the initial weight and $\rho : F \mapsto S$ is the final weight function that maps final states to elements in S.

Scope ambiguities, preferences of structures and degrees of grammaticalness are covered by linguistic criteria. With these criteria it is possible to compare analyses by means of scores. The linguistic criteria are formalized by the notion of a semiring (Didakowski 2007). Let $S \neq \emptyset$ be a set and $\oplus$ (called addition) and $\otimes$ (called multiplication) binary operations on S, then $(S, \oplus, \otimes, \bar{0}, \bar{1})$ is called a semiring if $(S, \oplus, \bar{0})$

is a commutative monoid, $(S, \otimes, \bar{1})$ is a monoid and $\otimes$ distributes over $\oplus$. Linguistic criteria are represented by this structure. To judge analyses via addition an additive idempotent semiring has to be used to create a partial order over S. Thus a partial order is defined by $(a \leq_S b) \Leftrightarrow (a \oplus b = a)$. Here $a \leq_S b$ means that a is "better" than b with respect to linguistic criteria.

It will be necessary to judge analyses by more than one linguistic criterion; therefore, the criteria are ranked by preference. To model this the *composition* of idempotent semirings is defined as follows (cf. Didakowski 2007): if a linguistic preference $(S_1, \oplus_1, \otimes_1, \bar{0}_1, \bar{1}_1) \succ (S_2, \oplus_2, \otimes_2, \bar{0}_2, \ \bar{1}_2) \succ ... \succ (S_n, \oplus_n, \otimes_n, \bar{0}_n, \bar{1}_n)$ is given and if for each semiring a partial order is defined by $\oplus$, then the composition $(S, \oplus, \otimes, \bar{0}, \bar{1}) = (S_1, \oplus_1, \otimes_1, \bar{0}_1, \bar{1}_1) \circ (S_2, \oplus_2, \otimes_2, \bar{0}_2, \bar{1}_2) \circ ... \circ (S_n, \oplus_n, \otimes_n, \bar{0}_n, \bar{1}_n)$ is the vectorization of the individual domains and of the operation $\otimes$. This corresponds to the *crossproduct* of semirings (cf. Hebisch and Weinert 1993). The operation $\oplus$ which compares analyses is defined in a special way, if $(a_1, a_2, ..., a_n) \in S$ and $(b_1, b_2, ..., b_n) \in S$ are given:

$$(a_1, a_2, ..., a_n) \oplus (b_1, b_2, ..., b_n) =$$

$$\begin{cases} (a_1, a_2, ..., a_n) & \text{if } (a_1, a_2, ..., a_n) = (b_1, b_2, ..., b_n) \\ (a_1, a_2, ..., a_n) & \text{if } a_1 = b_1 \text{ and } a_2 = b_2 \text{ and ... and } a_{k-1} = b_{k-1} \\ & \text{and } a_k \oplus_k b_k = a_k \\ & \text{with } k \leq n \text{ and } a_k \neq b_k \\ (b_1, b_2, ..., b_n) & \text{if } a_1 = b_1 \text{ and } a_2 = b_2 \text{ and ... and } a_{k-1} = b_{k-1} \\ & \text{and } a_k \oplus_k b_k = b_k \\ & \text{with } k \leq n \text{ and } a_k \neq b_k \end{cases} \tag{6.1}$$

The resulting semiring is now idempotent as well and a partial order can be defined by $\oplus$.

Extracting the most likely analyses with respect to linguistic criteria in a WFST T which is the result of the application of a cascade of analysis transducers is a classical best-path problem. Weights along a path of T are combined by the abstract multiplication and create costs. If several paths are in T their weight equals the abstract addition of weights of the different paths, that means the "best" cost (cf. Mohri 2002). The most likely analyses are simply represented by paths causing these "best" costs.

In the following the STTS (Stuttgart/Tübinger Tagset, a tagset for German) and the regular expression notation of Karttunen (1995) (slightly extended) are used.[1]

[1] See appendix for regular expression notation details. Here the precedence is defined top down. The distinction between the automaton A and the identity transducer that maps every string of A to itself is ignored.

3 Representation of the input and the analyses

In this section the representation of the input and the analyses are defined. The input consists of sequences of word boundaries, lemma forms and morphological properties. This sequences can be specified by a regular expression and represented compactly by an acyclic automaton (cf. Koskenniemi 1990). A concrete example of an input is given by the German sentence “Bill sieht den kleinen Hund” (Bill sees the little dog), whereas for lack of space only one reading is covered.[2]

```
{@@}
 {@}  Bill   {NE Case=nom}
 {@}  sehen  {VVFIN Number=sg}
 {@}  die    {ART Case=dat Number=pl Gender=fem}
 {@}  klein  {ADJA Case=nom Number=pl Gender=masc}
 {@}  Hund   {NN Case=nom Number=sg Gender=masc}
 {@}  .      {SYMBOL Type=punctuation}
{@@}
```

The tag @ marks a word boundary. Lemma forms (Bill, sehen , ...) and complex categories (`NE`, `VVFIN`, ...) are following them. Here the STTS is expanded by morphological features (`case`, `number`, etc.). The tag {`@@`} encloses the text marking an input block. Note, that the tag {`@@`} marks text segments, not sentences.

Like in Karlsson (1990) syntactic tags indicating dependency-oriented functions of words such as subjects, objects, modifier, etc. are used to mark dependency relations. But in contrast to the approach in Karlsson (1990) we use brackets instead of dependency markers (<,>) forming a constituent-like structure. In Karlsson (1990) dependency markers point toward the head of the dependency relation forming a constituent with implicit constituent borders. However, the representation is in many respects under-specified and avoids many difficult syntactic decisions concerning the linking of words. Here the bracketing groups the heads with their dependents achieving a more explicit dependency representation; within a grouping the dependents and the heads are explicitly defined by the syntactic functions. It is also possible to cover more than one hierarchical level of dependency order within a grouping. In such cases the linking of the words by the syntactic functions (without dependency markers) is definite.

One word in a dependency structure is independent and represents the head of the structure. The same holds for local dependency structures. Here we call the inde-

[2] The curly brackets denote possibly underspecified categories. The features are defined with respect to an inheritance hierarchy and are represented as transition labels. Underspecification is realized as the disjunction of all maximal subtypes of a super type.

pendent word a local structure ceiling. Via the local structure ceiling a local structure can be defined:

(1) A local dependency structure is the maximal subgraph of a local structure ceiling C which:

i.includes the word defining C

ii.does not contain any other local structure ceiling, and

iii.has connected pre-dependents and/or post-dependents

The local structure ceiling potentially can be a head or dependent in another dependency relation. Through this a local structure can be incorporated within a wider local structure via the local structure ceiling whereas the structure becomes a subpart of the wider structure. By this means a complex hierarchic dependency order together with a constituent-like structure can be built. The dependency structures are called local, because the structures should cover input fragments instead of the whole input text. Note that the input is not presented sentence-wise. The input example above is analyzed as follows:

```
{@@}
 %[
  {@}   Bill   {NE Case=nom}              {@SUBJ cei=no}
  {@}   sehen  {VVFIN Number=sg}          {@VMAIN cei=yes}
  %(
   {@}  die    {ART Case=dat Number=pl}   {@DET cei=no}
   {@}  klein  {ADJA Case=nom Number=pl}  {@ATTR cei=no}
   {@}  Hund   {NN Case=nom Number=sg}    {@OBJA cei=no}
  %)
  {@}   .      {SYMBOL Type=punctuation}
 %]
{@@}
```

In the example above the local dependency structure is marked by the brackets '[' and ']' and their internal phrase-like structure by the brackets '(' and ')'. The syntactic tags @VMAIN, @SUBJ and @OBJA mark the main verb, the subject and the accusative object of the sentence whereas the function @VMAIN also marks the local structure ceiling (cei=yes). The syntactic tag @DET marks the determiner and the syntactic tag @ATTR the noun attribute of the accusative object.

4 Generation of local structures

In our approach the constraints of a constraint grammar group words which are in a dependency relation and mark them by corresponding syntactic tags. In addition some constraints also mark unary relations (e.g. structures consisting of one word being the local structure ceiling). For the sake of readability of the following definitions the brackets '[' and ']' are denoted by the expressions $\mathtt{P}_\alpha$ and $\mathtt{S}_\alpha$ and the brackets '(' and ')' are denoted by $\mathtt{P}_\beta$ and $\mathtt{S}_\beta$. First, some help regular expressions are defined:

$$\texttt{NO\_LDS} =_{def} \sim\$[\mathtt{P}_\alpha \,|\, \mathtt{S}_\alpha] \tag{6.2}$$

$$\texttt{NON\_EMB} =_{def} [\texttt{NO\_LDS}\ \mathtt{P}_\alpha\ \texttt{NO\_LDS}\ \mathtt{S}_\alpha]^*\ \texttt{NO\_LDS} \tag{6.3}$$

$$\texttt{INCORP} =_{def} [\texttt{NO\_LDS}\ [\mathtt{P}_\alpha\texttt{.x.}\mathtt{P}_\beta]\ \texttt{NO\_LDS}\ [\mathtt{S}_\alpha\texttt{.x.}\mathtt{S}_\beta]]^*\ \texttt{NO\_LDS} \tag{6.4}$$

The expression NO_LDS does not allow any local dependency structure and the expression NON_EMB does not allow embedded local structures. The expression INCORP maps every local dependency structure to an incorporated structure of a wider local structure. With the help of these expressions the *grouping operator* which implements a constraint can be defined, if PATTERN denotes the local structure including the local structure ceiling with its pre/post-dependents:

$$\begin{gathered}\texttt{PATTERN (}\rightarrow\texttt{) \_} =_{def} \\ [\texttt{NON\_EMB}\ [\texttt{0.x.}\mathtt{P}_\alpha]\ [\texttt{PATTERN.o.INCORP}]\ [\texttt{0.x.}\mathtt{S}_\alpha]]^*\ \texttt{NON\_EMB}\end{gathered} \tag{6.5}$$

The operator achieves that a pattern PATTERN is optionally applied and bracketed. If an embedding is defined by the pattern, the corresponding local dependency structure is mapped to an incorporated structure. Within the pattern transductions can be performed that marks the local structure ceilings and their dependents and the type of dependency relation.

An example constraint (CONSTRAINT_1) that covers the sequence "a determiner is followed by a noun" is given. The expression LEMMA denotes all possible lemma forms:

$$\texttt{DET} =_{def} \texttt{\{@\} LEMMA \{ART\} [0.x.\{@DET cei=no\}]} \tag{6.6}$$

$$\texttt{NP\_HEAD} =_{def} \texttt{\{@\} LEMMA \{NN\} [0.x.\{@NP\_HEAD cei=yes\}]} \tag{6.7}$$

$$\texttt{CONSTRAINT\_1} =_{def} \texttt{DET NP\_HEAD (}\rightarrow\texttt{) \_} \tag{6.8}$$

Via epsilon the syntactic tags @DET and @NP_HEAD are introduced. Here the tag @NP_HEAD is an underspecified variant of @SUBJ, @OBJA, @GMOD, etc. and can be rewritten by other constraints. This approach will be exemplified by the following example constraint (CONSTRAINT_2). The expressions CLAUSE_START and CLAUSE_END denote potential beginnings end endings of clauses. We presume that these markers are inserted into the input:

$$\begin{gathered}\texttt{SUBJ} =_{def} \\ \mathtt{P}_\alpha\ \texttt{NO\_LDS [\{@NP\_HEAD cei=yes\}.x.\{@SUBJ cei=no\}] NO\_LDS}\ \mathtt{S}_\alpha\end{gathered} \tag{6.9}$$

$$\begin{gathered}\texttt{VMAIN} =_{def} \\ \texttt{\{@\} LEMMA \{VVFIN\} [0.x.\{@VMAIN cei=yes\}]}\end{gathered} \tag{6.10}$$

$$\begin{gathered}\texttt{CONSTRAINT\_2} =_{def} \\ \texttt{CLAUSE\_START SUBJ VMAIN CLAUSE\_END (}\rightarrow\texttt{) \_}\end{gathered} \tag{6.11}$$

The expression SUBJ denotes all local dependency structures with the local structure ceiling @NP_HEAD mapped to @SUBJ. The expression VMAIN denotes the main verb, the head of the grouping. The constraint covers main-clauses with the sequence “a subject followed by a main-verb”.

The constraints of a constraint grammar are combined by union. They compete in this definition, all constraints are checked in parallel:[3]

$$\begin{gathered}\texttt{CG} =_{def} \\ \texttt{PATTERN}_1\texttt{(}\rightarrow\texttt{)\_ | PATTERN}_2\texttt{(}\rightarrow\texttt{)\_ | ... | PATTERN}_n\texttt{(}\rightarrow\texttt{)\_}\end{gathered} \tag{6.12}$$

To build more complex dependency structures by incorporating of local dependency structures to wider local dependency structures a constraint grammar is composed iteratively. The iteration is defined as follows, if i denotes the maximum iteration and $i > 1$:

$$\begin{aligned}\texttt{CG}^i &=_{def} \texttt{CG}^{i-1} \texttt{ .o. CG} \\ \texttt{CG}^1 &=_{def} \texttt{CG}\end{aligned} \tag{6.13}$$

Here the maximum depth of incorporations and the maximum depth of the hierarchic order is predefined by i. Like in finite state intersection grammars an adequate constraint grammar generally can not be compiled into one monolithic transducer. In practice, this would be too time and space consuming (cp. Tapanainen 1997). So the subparts of a constraint grammar are applied sequentially to the input.

The most important property of constraint grammar constraints is the ability to disambiguate. However, the definition of the constraints in this section does not allow any disambiguation yet. If a constraint grammar is applied to an input the resulting analysis contains the desired analyses but many others as well (note that the constraints are defined as optional). This problem is discussed in section 5.

5 Disambiguation via linguistic criteria

In our approach we use scores to implement the disambiguation property of constraints realizing the preference of syntactic structures and degrees of grammaticalness. Here we use two strategies: longest match and binding force of dependents and heads.

[3] This definition is comparable to the construction of a syntactic dictionary in Roche (1997).

To implement a longest match as a preference strategy we follow the approach in Didakowski (2007). There the strategy is formulated by the two linguistic criteria chunk inclusiveness and chunk connectedness. Both are formalized by the tropical semiring $(\mathbb{R} \cup \{\infty\}, min, +, \infty, 0)$ and are ranked as follows: chunk inclusiveness $\succ$ chunk connectedness. The approach is outlined shortly. Material within brackets (that group dependents and heads) get "good" costs with respect to chunk inclusiveness. By this an exhaustive grouping is achieved. In addition to that the grouping bracket pairs get "bad" costs with respect to chunk connectedness to keep the structures together. To implement this method the weights have to be assigned by corresponding filters within the grouping operator. An example is given by the input fragment "weil der Mann schläft" (because the Man is sleeping). Words within brackets get a cost -1 and bracket pairs a cost 1 with respect to the individual linguistic criteria. In the following example possible analyses are ordered by their degree of acceptance:

$\langle -2, 1 \rangle$ weil [der$_{\texttt{@DET}}$ Mann$_{\texttt{@NP\_HEAD}}$] schläft

$\langle -2, 2 \rangle$ weil [der$_{\texttt{@NP\_HEAD}}$] [Mann$_{\texttt{@NP\_HEAD}}$] schläft

$\langle 0, 0 \rangle$ weil der Mann schläft

The binding force of dependent and head in a dependency relation is formalized by the *constraint optimization criterion*: dependency structures receiving the best binding scores by constraints are preferred. The criterion is formalized by the max-semiring $(\mathbb{R} \cup \{-\infty\}, max, +, -\infty, 0)$. The scores are coded by numbers, negative numbers punish dependency relations gradually (for example if the gender, number or case agreement is violated) and positive numbers support dependency relations gradually. The scores are directly assigned within the pattern of the grouping operator. Here the constraint optimization criterion has priority over the longest match. That is the longest match is simply a default strategy. So the ordering of the criteria is defined as follows: constraint optimization $\succ$ chunk inclusiveness $\succ$ chunk connectedness.

An example concerning the preference of syntactic structures is given. A constraint grammar is assumed analyzing the input "Peter verkauft das Kleid der Frau." (Peter sells the dress to the woman. / Peter sells the woman's dress.) without scoring with respect to the constraint optimization criterion. Now among others the following structures would be generated and ordered by the linguistic criteria:[4]

$\langle 0, -13, 4 \rangle$[[Peter$_{\texttt{@SUBJ}}$] verkauft$_{\texttt{@MAINV}}$ [das$_{\texttt{@DET}}$ Kleid$_{\texttt{@AOBJ}}$ [der$_{\texttt{@DET}}$ Frau$_{\texttt{@GMOD}}$]].]

$\langle 0, -11, 4 \rangle$[[Peter$_{\texttt{@SUBJ}}$] verkauft$_{\texttt{@MAINV}}$ [das$_{\texttt{@DET}}$ Kleid$_{\texttt{@AOBJ}}$][der$_{\texttt{@DET}}$ Frau$_{\texttt{@DOBJ}}$].]

[4] In case of incorporations the scores are abstractly multiplied.

In the example sentence the nominal phrase “der Frau” (the woman’s) is a genitive attribute (GMOD) of “das Kleid” (the dress) or a dative object (@DOBJ) of the main verb. Longest match prefers the reading with the genitive attribute. If the reading with the dative object should be preferred even this can be achieved via the constraint optimization criterion. Here it is assumed, that the binding force of the main verb and of the dative object is 5:

$$\langle 5,-11,4\rangle[[\text{Peter}_{\texttt{@SUBJ}}]\ \text{verkauft}_{\texttt{@MAINV}}\ [\text{das}_{\texttt{@DET}}\ \text{Kleid}_{\texttt{@AOBJ}}][\text{der}_{\texttt{@DET}}\ \text{Frau}_{\texttt{@DOBJ}}].]$$

$$\langle 0,-13,4\rangle[[\text{Peter}_{\texttt{@SUBJ}}]\ \text{verkauft}_{\texttt{@MAINV}}\ [\text{das}_{\texttt{@DET}}\ \text{Kleid}_{\texttt{@AOBJ}}\ [\text{der}_{\texttt{@DET}}\ \text{Frau}_{\texttt{@GMOD}}]].]$$

Sometimes it is desired to outrank the constraint optimization criterion by a longest match strategy. Assuming that the violation of agreement is heavily punished a sentence can’t be analyzed in which any agreement is violated. However, we still want to get an analysis for such sentences. This can be realized by a longest match constraint which has priority over the constraint optimization criterion: chunk inclusiveness $\succ$ chunk connectedness $\succ$ constraint optimization $\succ$ chunk inclusiveness $\succ$ chunk connectedness. The weights according to the higher ranked longest match constraint are exclusively assigned with respect to the corresponding constraints.

6 Avoiding redundancies

If a constraint grammar as defined in section 4 is applied to an input redundancies occur. The same structures can be generated several times during the composition of the analysis transducers of a constraint grammar. The problem is caused by the circumstance that an analysis transducer does not know whether a structure has already been generated by a previous step. This is overcome by keeping track of the new structures which have been inserted by a previous step. The new structures have to be part of the structures generated by the following step. By this, it can be is guaranteed that a structure is never generated more than once.

To keep track of the new generated structures we rewrite the brackets of local structures that are not incorporated in a new generated structure: the bracket $\texttt{P}_\alpha$ is rewritten by $\texttt{P}_\gamma$ and the bracket $\texttt{S}_\alpha$ by $\texttt{S}_\gamma$. In this context the brackets $\texttt{P}_\gamma$ and $\texttt{S}_\gamma$ have to be handled in conjunction with embedded structures. To realize this, the help regular expressions presented in section 4 are adjusted as follows:

$$\texttt{NO\_LDS} =_{def} \sim\texttt{\$[P}_\alpha\texttt{|S}_\alpha\texttt{|P}_\gamma\texttt{|S}_\gamma\texttt{]} \tag{6.14}$$

$$\texttt{NON\_EMB} =_{def} \texttt{[NO\_LDS [[P}_\alpha\texttt{.x.P}_\gamma\texttt{]|P}_\gamma\texttt{] NO\_LDS [[S}_\alpha\texttt{.x.S}_\gamma\texttt{]|S}_\gamma\texttt{]]}^*\ \texttt{NO\_LDS} \tag{6.15}$$

$$\texttt{INCORP} =_{def} \texttt{[NO\_LDS [[P}_\alpha\texttt{|P}_\gamma\texttt{].x.P}_\beta\texttt{] NO\_LDS [[S}_\alpha\texttt{|S}_\gamma\texttt{].x.S}_\beta\texttt{]]}^*\ \texttt{NO\_LDS} \tag{6.16}$$

Additionally the following regular expression is defined to ask for a new generated structure:

$$\texttt{NEW\_ST} =_{def} \texttt{?* P}_\alpha\ \texttt{[\textasciitilde\$[P}_\alpha\texttt{|S}_\alpha\texttt{]] S}_\alpha\ \texttt{?*} \tag{6.17}$$

On the basis of the regular expressions defined above the *incorporating grouping operator* can be defined:

$$\begin{gathered}\texttt{PATTERN} \ (\rightarrow)\ \texttt{\_()\_} =_{def}\\ \texttt{[NON\_EMB[0.x.P}_\alpha\texttt{][NEW\_ST.o.PATTERN.o.INCORP][0.x.S}_\alpha\texttt{]]}^*\texttt{NON\_EMB}\end{gathered} \tag{6.18}$$

By the grouping operator and the incorporating grouping operator two types of constraint grammar can be defined: a) a constraint grammar as defined in section 4 – this variant is denoted by CG'; b) a constraint grammar that is defined by the incorporating grouping operator – this variant is denoted by CG.

$$\begin{gathered}\texttt{CG'} =_{def}\\ \texttt{PATTERN}_1(\rightarrow)\texttt{\_} \ | \ \texttt{PATTERN}_2(\rightarrow)\texttt{\_} \ | \ \dots \ | \ \texttt{PATTERN}_n(\rightarrow)\texttt{\_}\end{gathered} \tag{6.19}$$

$$\begin{gathered}\texttt{CG} =_{def}\\ \texttt{PATTERN}_1(\rightarrow)\texttt{\_()\_} \ | \ \texttt{PATTERN}_2(\rightarrow)\texttt{\_()\_} \ | \ \dots \ | \ \texttt{PATTERN}_n(\rightarrow)\texttt{\_()\_}\end{gathered} \tag{6.20}$$

To build more complex dependency structures by incorporation of local dependency structures to wider local dependency structures, the iteration of a constraint grammar is defined as follows, if i denotes the maximum iteration and $i > 1$:

$$\begin{aligned}\texttt{CG}^i &=_{def} \texttt{CG}^{i-1}\ \texttt{.o. CG}\\ \texttt{CG}^1 &=_{def} \texttt{CG'}\end{aligned} \tag{6.21}$$

Unlike the iteration defined in section 4 redundant structures are avoided by this definition.

7 Detecting the maximum depth of dependency structures

To realize incorporations a constraint grammar is applied iteratively to the input. Here it is possible, that much extra work is done by the applications of a constraint grammar that does not introduce further structures into the input. Thus, the maximum depth of dependency structures should be detected. This is done in accordant to Roche (1997) by an unbound iteration of the application of a constraint grammar which is denoted by $\texttt{CG}^\infty$. The break condition is simply the test of equal size of the result of an application and the result of an application of a previous step. That is it is checked whether new structures are generated or not. Note that each structure is generated only once, guaranteeing the halt of the algorithm. This parsing algorithm is given in figure 6.22.

```
Input:                 INPUT, CG, CG'                                                    (6.22)

ANALYSE1 =_def         INPUT.o.CG'

while                  ( |ANALYSE1| ≠ |ANALYSE2 =_def ANALYSE1.o.CG| )
                           ANALYSE1 =_def ANALYSE2

Output:                ANALYSE1
```

8 Testing and results

The presented approach is implemented by the Syntactic Constraint Parser (SynCoP), which is based on WFSTs (cf. Didakowski 2007; Geyken et al. 2008). SynCoP consists of a grammar compiler, a grammar-driven parser, and a preprocessing module which comprises tokenizing and recognition of multi-word units. The engine admits specification of the parser along with the preprocessing module by means of a grammar written in XML. Thus the engine can be easily adapted to individual conceptions of analysis.

A German test grammar was created via hand-written constraints. To create a more modular, more intuitive and more efficient constraint grammar the grammar is split up into several levels of processing: (1) a constraint grammar for constituents; (2) a constraint grammar for sub-clauses; (3) a constraint grammar for main-clauses.

$$\mathtt{CG} =_{def} \mathtt{CG}^{\infty}_{constituent} \;\; \mathtt{.o.} \;\; \mathtt{CG}^{1}_{sub-clause} \;\; \mathtt{.o.} \;\; \mathtt{CG}^{1}_{main-clause} \tag{6.23}$$

The handling of congruency is covered by our grammar. The linking of sub-clauses is not embedded. As Left embeddings or right embeddings are replaced by iteration the constraint grammar for sub-clauses is applied only once. Note that deeper central embeddings are very rare. This is a common approach in fully automatic corpus annotation (cf. Koskenniemi 1990). The size of the compiled sub-constraint grammars are shown in the following table:

grammar	states	transitions
CG'$_{constituent}$	65127	630324
CG$_{constituent}$	91220	1082137
CG$_{sub-clause}$	7606	677379
CG$_{main-clause}$	18901	1489968

(6.24)

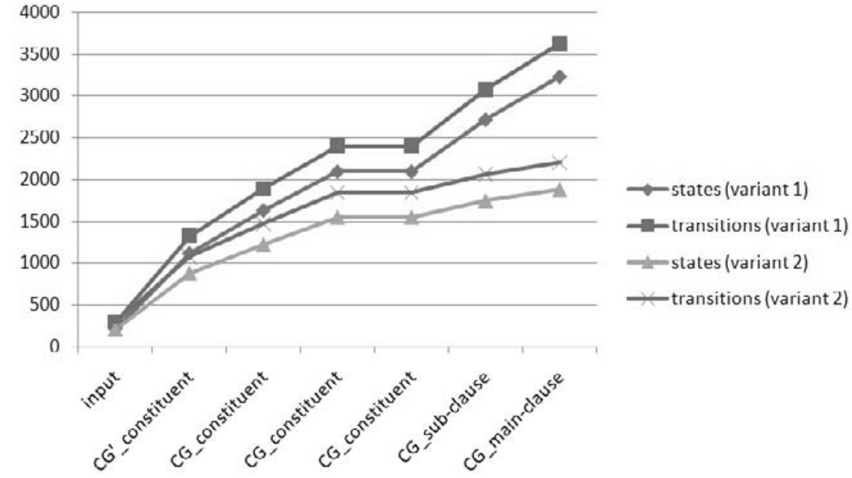


Figure 1. Number of states and transitions in respect of the level of analysis

During the analysis by our constraint grammar the resulting WFST is growing level by level before the most likely syntactic readings are extracted by a best-path search. This is shown by the analysis of the German sentence *Die Frau, die das Auto auf dem Schrottplatz sehen wird, weint. (The woman, who is going to see the car at the wasteyard, is crying).*[5] [6] The steps of this analysis concerning the number of states and transitions are shown in figure 1. Here, the number of states and transitions concerning the default application of the constraint grammar is shown by variant 1. The input WFST has 208 states and 290 transitions. The final result has 3234 states and 3632 transitions. Here the amount of states and transitions of the result transducer is growing quite slow. Problems of large intermediate results as reported in Tapanainen (1997) do not occur. So it is possible to analyze larger input blocks without computational problems.

To speed up the analysis a structure packing strategy is applied. Here we follow the idea of Yli-Jyrä (2004) of the simplification of weighted automata. Our strategy is based on the fact that finite state automata are closed under substitution. The idea is that material within groupings which does not belong to the chunk ceiling is packed and replaced by an index. After parsing the indexes are replaced by the corresponding material. This strategy is shown in figure 1 by variant 2. With this strategy the final

[5] PP-attachment and the linking of sub-clauses is not covered by our constraint grammar yet.

[6] This sentence is analyzed as follows: [[Die$_{@DET}$ Frau$_{@SUBJ}$], [die$_{@SUBJ}$ [das$_{@DET}$ Auto$_{@DOBJ}$] [auf$_{@PP\_HEAD}$ [dem$_{@DET}$ Schrottplatz$_{@P}$]] sehen$_{@VMAIN}$ wird$_{@VAUX}$], weint$_{@VMAIN}$.]

result has 1879 states and 2203 transitions. Here the amount of states and transitions shows a slower increase.

9 Conclusion and future work

A new approach to fully automatic annotation of large corpora with weighted finite state transducers was presented. A realization of a robust annotation by local syntactic tagging is proposed. Linguistic criteria make it possible to model preferences of syntactic readings and degrees of grammaticalness. The analysis system SynCoP implements the represented approach by a handwritten grammar.

For future grammar implementations syntactic annotated corpora should be considered. The patterns of the constraints and the binding force should be extracted from corpora. Linguistic generalizations like case, gender and number agreement should be handled by special filters.[7] The parser will be used in the construction of the German syntactic collocation database *Word Profile* (cf. Geyken et al. 2008).

10 Appendix:Notations

$\sim$A	complement
$A	contains
A*	Kleene star
A B	concatenation
A | B	union
A & B	intersection
A .x. B	crossproduct
A .o. B	composition
$\{$A feat$_1$=value$_{1_x}$...feat$_n$=value$_{n_x}$$\}$	category A with specified features
feat$_x$=value$_{x_1}$:value$_{x_2}$	mapping of feature values
$\langle \omega \rangle$	weights
[and]	grouping of expressions
?	sigma
?*	sigma star
0	epsilon

[7] An interesting approch is presented in Tapanainen and Järvinen (1994): rule-based elements are connected with corpus-based patterns in the context of the constraint grammar formalism (Karlsson 1990). However, there is a strict separation of the rule-based parser and the corpus-based parser.

References

Abney, Steven (1990). Syntactic Affixation and Performance Structures. In Denis Bouchard and Katherine Leffel (eds.), *Views on Phrase Structure*, Kluwer.

Didakowski, Jörg (2007). SynCoP - Combining Syntactig Tagging with Chunking Using Weighted Finite State Transducers. In *Proceedings of FSMNLP 2007*.

Geyken, Alexander, Jörg Didakowski, and Alexander Siebert (2008). Generation of Word Profiles on the Basis of a Large Balanced German Corpus. In *Proceedings of EURALEX*.

Hebisch, Udo and Hanns J. Weinert (1993). *Halbringe*. Stuttgart:Teubner.

Karlsson, Fred (1990). Constraint Grammar as a Framework for Parsing Running Text. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, volume 3, 168–173.

Karlsson, Fred, Atro Voutilainen, Juha Heikkilä, and Arto Anttila (1995). *Constraint Grammar - A Language-Independent System for Parsing Unrestricted Text*. Berlin/New York: Mouton de Gruyter.

Karttunen, Lauri (1995). The Replace Operator. In *Meeting of the Association for Computational Linguistics*, 16–23.

Kilgarriff, A., P. Rychly, P. Smrz, and D. Tugwell (2004). The Sketch Engine. In *Proceedings of the Eleventh EURALEX International Congress*, 105–116.

Koskenniemi, Kimmo (1990). Finite-State Parsing and Disambiguation. In *Proceedings of the the 13th International Conference on Computational Linguistics (COLING 90)*, volume 2, 229–232.

Mohri, Mehryar (2002). Semiring Frameworks and Algorithms for Shortest-Distance Problems. *Languages and Combinatorics* 7(3):321–350.

Roche, Emmanuel (1997). *Parsing with Finite State Transducers*, 241–281. MIT Press.

Sinclair, John (1991). *Corpus, Concordance, Collocation*. Oxford: Oxford University Press.

Tapanainen, Pasi (1997). *Applying a Finite-State Intersection Grammar*, 311–327. MIT Press.

Tapanainen, Pasi (1999). *Parsing in two Frameworks: Finite-State and Functional Dependency Grammar*. University of Helsinki.

Tapanainen, Pasi and Timo Järvinen (1994). Syntactic Analysis of Natural Language Using Linguistic Rules and Corpus-Based Patterns. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, 629–634.

Voutilainen, Atro (1997). *Designing a (Finite-State) Parsing Grammar*, 283–310. MIT Press.

Yli-Jyrä, Anssi (2004). Simplification of Intermediate Results during Intersection of Multiple Weighted Automata. In *Weighted Automata: Theory and Applications*, 46–48, Technische Universität Dresden.

Part II
Extraction of lexical knowledge from text resources

Towards improved text understanding with WordNet

Christiane Fellbaum, Peter Clark and Jerry Hobbs

Abstract. WordNet could be described as a light-weight knowledge base, where information about words and the concepts they express comes largely from a limited number of semantic relations. Currently, WordNet encodes only a very small part of the knowledge that people access when processing texts. We report on recent enhancements to WordNet's contents that maintain and exploit its structure while moving closer towards the goal of deep language understanding. The enhancements are targeted so as to capture the kind of linguistic and world knowledge that is shared among speakers and hence tends to remain unexpressed in naturalistic text. We discuss the addition of formalized synset definitions (glosses), the typing of morphosemantic links among nouns and verbs, and the encoding of a small number of "core theories" about the most commonly used concepts.

Introduction

People routinely derive knowledge from texts that is not expressed on the surface. For examples, given a sentence like

(1) A soldier was killed in a gun battle

people make a number of inferences, including the following:

(2a) Soldiers were fighting one another

(2b) The soldiers had guns with live ammunition

(2c) Multiple shots were fired

(2d) One soldier shot another soldier

(2e) The shot soldier died as a result of injuries caused by the shot(s)

(2f) The time interval between the fatal shot and the soldier's death was short

(2f) is particularly difficult to capture; while the soldier may have died after the end of the battle, his death could not come, say, weeks later in a hospital or rehabilitation home.

Constructing a complex scenario on the basis of sparse explicit information is easy for humans, who possess a wealth of world knowledge that accompanies lexical, or word, knowledge. But trying to explicate this knowledge and encode it for use by automatic systems remains an elusive goal. Knowledge bases like Cyc (Lena and Guha 1990) attempt to manually encode, in machine-tractable format, tens of thousands of statements ("microtheories") reflecting the kind of common sense and world knowledge speakers access when processing sentences like (1). The task seems open-ended and remains a major challenge.

WordNet (Fellbaum 1998a,b; Miller 1995) presents one avenue for making inroads into the challenge of knowledge representation. WordNet already has broad coverage, multiple lexico-semantic connections, and encodes significant knowledge (albeit informally) in its glosses and example sentences. It can thus be viewed as an extensively leverageable resource with significant potential for reasoning. In fact, WordNet already plays a central role in many question-answering systems. For example, twenty-one of the twenty-six teams in the recent PASCAL RTE3 challenge used WordNet (Giampiccolo et al. 2007), and without it or a similar resource, systems are largely restricted to simple word rearrangement inferences. We are developing several augmentations to WordNet to improve its utility further, and we report here on our experiences to date.

We are performing experiments with recognizing textual entailment (RTE), i.e., determining whether a hypothesis sentence H follows from some text T, such as in the pair below:

(H) A soldier was killed in a gun battle

(T) A soldier died

Many existing RTE systems (e.g., Adams et al. 2007; Chambers et al. 2007) largely work by statistically scoring the match between T and H, but this arguably sidesteps the kind of deep language understanding that requires building a coherent, internal representation of the overall scenario the input text was intended to convey.

We are developing and testing our work with Boeing's suite of language processing tools. We describe how an initial interpretation of text is obtained using these and how subsumption between texts is computed. We then describe our experience working with several recent WordNet enhancements, with the aim of providing some preliminary insights into a ways for creating and leveraging world knowledge in WordNet to achieve deeper language understanding.

As an experimental test bed we have developed an RTE-style test suite of 250 Text-Hypothesis pairs, where in half the pairs H is entailed by T and in the other half H is not entailed. As our goal is deeper semantic processing, the texts are syntactically simpler than the PASCAL RTE sets (`www.pascal-network.org`) but semantically challenging to process. The suite is publicly available (Clark et al. 2008c), along with an analysis of the types of world knowledge required for each pair. The examples in this paper come from our test suite.

Text interpretation and subsumption

We briefly describe our approach to interpreting text. Parsing is performed using SAPIR, a mature, bottom-up, broad coverage chart parser (Harrison and Maxwell 1986). The parser's cost function is biased by a database of manually and corpus-derived "tuples" (good parse fragments), as well as hand-coded preference rules. During parsing, the system also generates a logical form (LF), a semi-formal structure between a parse and full logic, loosely based on (Schubert and Hwang 2000). The LF is a simplified and normalized tree structure with logic-type elements, generated by rules parallel to the grammar rules, that contains variables for noun phrases and additional expressions for other sentence constituents. Syntactic parsing and part of speech tagging are performed at this stage while semantic analysis such as word sense discrimination and semantic role assignment are deferred, and there is no explicit quantifier scoping. The LF is used to generate ground logical assertions. For details see Clark et al. (2008a,b).

A basic operation for reasoning is determining if one set of clauses subsumes another. For example, the (the logic for) *A person likes a person* subsumes *A man loves a woman*. This basic operation is used both to determine if an axiom applies, and, in RTE, to determine whether a clause H subsumes a text T or its axiom-expanded elaboration. A set S_1 of clauses subsumes another S_2 if each clause in S_1 subsumes some (different) member of S_2. A clause C_1 subsumes another C_2 if both arguments (for binary predicates) of C_1 subsume the corresponding arguments in C_2, and the predicates of C_1 and C_2 "match." An argument A_1 subsumes another argument A_2 if a sense of A_1 is a hypernym of a sense of A_2 (thus considering all word senses of A_1 and A_2). We implement rules for predicate matching (Clark et al. 2008a,b).

Language allows us to express a given proposition in different ways. An event can be encoded as a verb, a deverbal noun or a resultative adjective, as is *X destroyed the city* vs. *The destruction of the city (by X)* vs. *the destroyed city*. To handle these variants involving different parts of speech (POS), our system considers all POS when finding the word senses of a word, independent of its POS in the original text. Combined with the predicate-matching rules, we have a powerful way of aligning expressions involving different POS.

Use of WordNet's glosses: translation to logic

WordNet encodes primarily paradigmatic relations among words and synsets. Only the definitions ("glosses") and example phrases that are part of each synset give information about which words the synset members can co-occur with. In addition, the glosses contain substantial amounts of world knowledge that could be useful for the semantic interpretation of text, and we have been exploring leveraging them by translating them into first-order logic.[1]

To perform the translation, each gloss was converted into a sentence of the form "word is gloss" and parsed with the Charniak parser. To translate the parse tree into its LF, a system called LFToolkit was used.[2] Lexical items are translated into LF fragments involving variables. As syntactic relations are recognized, variables contained in the constituents are identified with one other. For example, in *John works*, the word *John* tells us that there is an x_1 such that John(x_1). The word *works* tells us that there is an x_2 and an e such that e is a working event by x_2 and e is in the present: work(e,x_2) and present(e). When *John* is recognized as the subject of *works*, the variables x_1 x_2 are set equal to each other.

Such rules were developed for a large number of English syntactic constructions. The modified WordNet glosses could then be translated into LF, which were subsequently converted into axioms. Predicates are assigned word senses based on the recently released WordNet sense-tagged gloss corpus; this corpus was constructed by manually linking many tokens in the glosses with specific synsets (Clark et al. 2008a,b).

The translation was applied to the more than 110,000 WordNet glosses, with particular attention to glosses for the 5000 synsets in CoreWordNet (Boyd-Graber et al. 2006). It resulted in good translations for 59.4% of the 5000 core glosses, with lower quality for the entire gloss corpus. Where there was a failure, it was generally the result of a faulty parse; the glosses included definition-specific constructions for which no LFToolkit rules had been written. For some such these cases, the constituents of the construction are translated into logical form, so that no information is lost; what is lost is the binding between variables that provides the connections among the constituents. For instance, in the *John works* example, we would know that there was someone named *John* and that somebody *works*, but we would not know that they were coreferent. Altogether 98.1% of the 5000 core glosses were translated into correct axioms or axioms that had all the propositional content. [3]

Using the glosses for text understanding

We used a combination of the logicalized glosses and those from XWN to infer implicit information from text. Although the quality of the logic is generally poor

[1] We have also experimented with Extended WordNet (XWN), a similar database constructed several years ago by Moldovan and Rus (2001), but based on automatically disambiguated glosses and some modifications to the WordNet database.

[2] LFToolkit was developed by Nishit Rathod at ISI.

[3] The remaining 1.9% of these glosses misparsed due to unresolvable structural ambiguities.

(for a variety of reasons largely attributable to the fact that the glosses were never written for machine processing), our software was able to infer conclusions that help evaluate hypotheses. Consider an example:

(T) Britain puts curbs on immigrant labor from Bulgaria and Romania

(H) Britain restricted workers from Bulgaria

The system uses the definition of the verb synsets restrict, restrain, *place limits on* as well as WordNet's knowledge *put* and *place* and *curb* and *limit* are synonyms, respectively; additionally, WordNet tells one that a *laborer* is a *worker*. In our experiments, the glosses were used to answer 5 of the 250 entailment questions, four of them correctly.

More commonly, the glosses came tantalizingly close to providing the needed knowledge but failed ultimately because of a missing link between related words. This experience confirmed that WordNet would be even more valuable if its words and synsets were more densely connected.

Typing morphosemantic links

Earlier versions of WordNet contained "morphosemantic" links among semantically related words from different parts of speech, where one was morphologically derived from another (Fellbaum and Miller 2003). An example is is the cluster *employ-employee-employ-employment*, where each word shares a common core sense. The approximately 210,000 morphosemantic links turn out to be particular important for mapping between verbs and deverbal nominals like *destroy* and *destruction*, which frequently occur in Texts and Hypotheses. However, the links do not state the type the semantic relation; Wordnet does not tell one that *employee* is the UNDERGOER of an *employ* event, nor that *employment* is the *employ* EVENT or RESULT itself, nor that the *employer* is the AGENT performing the *employing* event. This limits WordNet's ability to support semantic role labeling. In addition, not being able to distinguish the semantics of the relations can cause errors in reasoning, and fail to distinguish between entailed and non-entailed hypotheses:

(T) Detroit produces fast cars.

(H1) Detroit's product is fast.

(H2) *Detroit's production is fast. [NOT entailed]

(T) The Zoopraxiscope was invented by Mulbridge.

(H1) Mulbridge was the inventor of the Zoopraxiscope.

(H2) *Mulbridge was the invention of the Zoopraxiscope. [NOT entailed]

We typed the noun-verb links, using a semi-automatic process. First, the computer makes a guess at the default semantic relation based on the morphological relation between the noun and the verb (e.g., nouns derived from verbs via emph-er affixation usually denote agents) and the location of the two synsets in WordNet's taxonomy (the lexicographer files, labeled *noun.person, verb.communication* etc.). Second, a human manually validates and corrects these automatically compiled pairs — a considerably faster progress than encoding them from scratch. We settled on a relatively small inventory of semantic relations in an attempt to avid overly fine-grained distinctions but capture significant relations:

- Agent (employ-employer)
- Undergoer/Patient (employ-employee)
- Instrument (shred-shredder)
- Recipient (grant-grantee)
- Event (employ-employment)
- Result (produce-product)
- Body Part (adduct-adductor)
- Vehicle (cruise-cruiser)
- Location (plant-planter)

The resulting database of 21,000 typed links constitutes a major addition to WordNet in support of deep language processing. One of the surprising side results of this effort was discovering how often the "default" relations do not apply (Fellbaum et al. 2007).

Core theories

While WordNet's glosses and links contain world knowledge about specific entities and relations among them, they do not capture more fundamental knowledge about language and the world, for example knowledge about space, time, and causality. Such knowledge is essential for understanding many types of text but unlikely to be expressed in dictionary definitions, nor can such knowledge be acquired automatically. To address the need for basic world knowledge and support deeper reasoning, we manually encoded a number of theories in the style of lexical decomposition. We axiomatized abstract core theories that underlie the way we talk about events and event structure (Hobbs 2008). Among these are theories of composite entities (things made of other things), scalar notions (of which space, time, and number are specializations), change of state, and causality. For example, in the theory of change of state, the predication change(e_1,e_2) says there that is a change of state from state e_1 to state e_2. The predication changeFrom(e_1) says there is a change out of state e_1. The predication changeTo(e_2) says there is a change into state e_2. An inference from changeFrom(e_1) is that e_1 no longer holds. An inference from changeTo(e_2) is that e_2 now does hold. In the theory of causality (Hobbs 2005), the predication cause(e_1,e_2), for e_1 causes e_2, is explicated. One associated inference is that if the causing happens, then the effect e_2 happens.

We are connecting the core theories with WordNet by mapping the 5,000 core WordNet synsets to the theory predicates. For example, there are 450 synsets having to do with events and event structure, and we encode their meanings in terms of core theory predicates. For example, if x lets e happen, then x does not cause e not to happen.

WordNet includes a sense of *go* that is *changeTo*, as in *he went wild*: $go(x,e) \leftrightarrow changeTo(e)$

where x is the grammatical subject of the eventuality e. If x frees y, then x causes a change to y being free:

$free(x,y) \leftrightarrow cause(x,changeTo(free(y)))$

Given the mappings and the core theories themselves, this is enough to answer the following entailment pair

(T) The captors freed the hostage

(H) The captors let the hostage go free

via:

- let(x,go(y,free(y)))
- $\leftrightarrow not(cause(x,not(changeTo(free(y)))))$
- $\leftrightarrow cause(x,changeTo(free(y)))$
- $\leftrightarrow free(x,y)$

Examples like these indicate the potential usefulness of our current developments for text inference.

Preliminary evaluation

The overall score on the 250-sentence test suite stands at 61.2% correct. The typing of WordNet's morphosemantic links in particular appears to make positive contributions. We also ran our software on the PASCAL RTE3 dataset (Giampiccolo et al. 2007), scoring 55.7% (this excludes cases where no initial LF could be constructed due to parse/LF generation failures).[4].

Conclusion

People effortlessly process language at a deep level and construct a coherent representation of the scene the speaker or writer intends to convey. To perform such understanding automatically, remains a considerable challenge that requires access to large amounts of explicit world knowledge. We described our work in progress, focusing on various enhancements of WordNet, and some initial experiences with these augmentations. WordNet already provides extensive leverage for language processing, as evidenced by the many researchers that use it; our paper contributes some preliminary insight into directions for further developing this resource. Although limited experience suggests that the enhancements carry promise for further improving deep language processing.

Acknowledgment

This work was performed under the DTO AQUAINT program, contract N61339-06-C-0160.

References

Adams, Rod, Gabriel Nicolae, Cristina Nicolae, and Sanda Harabagiu (2007). Textual Entailment Through Extended Lexical Overlap and Lexico-Semantic Matching. In *Proceedings of the ACL-PASCAL Workshop on Textual and Entailment and Paraphrasing*, 119–124.

Boyd-Graber, Jordan, Christaine Fellbaum, Daniel Osherson, and Robert Schapire (2006). Adding Dense, Weighted, Connections to WordNet. In *Proceedings of the Third Global WordNet Meeting*, Jeju Island, Korea.

Chambers, Nathanael, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and Christopher D. Manning (2007). Learning Alignments and Leveraging Natural Logic. In *Proceedings of the ACL-PASCAL Workshop on Textual and Entailment and Paraphrasing*, 165–170.

[4] For details on the evaluation see Clark et al. (2008a,b)

Clark, Peter, Christiane Fellbaum, and Jerry Hobbs (2008a). Using and Extending WordNet to Support Question Answering. In A. Tanacs, D. Csendes, V. Vincze, C. Fellbaum, and P. Vossen (eds.), *Proceedings of the Fourth Global WordNet Conference*, 111–119, University of Szeged, Hungary.

Clark, Peter, Christiane Fellbaum, Jerry Hobbs, Phil Harrison, William Murray, and John Thompson (2008b). Augmenting Wordnet for Deep Understanding of Text. In *ACL-SigSem*, Venice, Italy.

Clark, Peter, Jerry Hobbs, and Christiane Fellbaum (2008c). The BPI Entailment Test Suite. http://www.cs.utexas.edu/∼pclark/bpi-test-suite/.

Fellbaum, Christiane (1998a). *WordNet*. Cambridge: MIT Press.

Fellbaum, Christiane (1998b). *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

Fellbaum, Christiane and George A. Miller (2003). Morphosemantic Links in WordNet. *Traitement Automatique des Langues* 44(2):69–80.

Fellbaum, Christiane, Anne Osherson, and Peter Clark (2007). Putting Semantics into WordNet's "Morphosemantic" Links. In *Proceedings of the Third Language and Technology Conference*, Poznan, Poland.

Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and Bill Dolan (2007). The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL Workshop on Textual and Entailment and Paraphrasing*, 1–9.

Harrison, Philip and Michael Maxwell (1986). A New Implementation of GPSG. In *Proceedings of the sixth Canadian Conference on AI*, 78–83.

Hobbs, Jerry (2005). Toward a Useful Notion of Causality for Lexical Semantics. *Journal of Semantics* 22:181–209.

Hobbs, Jerry (2008). Encoding Commonsense Knowledge.

Lena, Douglas B. and Ramanathan V. Guha (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Reading: Addison-Wesley.

Miller, George A. (1995). Wordnet: A Lexical Database for English. *Communications of the ACM* 38(11):39–41.

Schubert, Lenhart K. and Chung Hee Hwang (2000). Episodic Logic Meets Little Red Riding Hood: A Comprehensive, Natural Representation for Language Understanding. In L. Iwanska and S. C. Shapiro (eds.), *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, 111–174, Menlo Park and Cambridge: MIT/AAAI Press.

Influence of accurate compound noun splitting on bilingual vocabulary extraction

Marcin Junczys-Dowmunt

Abstract. The influence of compound noun splitting on a German-Polish bilingual vocabulary extraction task is investigated. To accomplish this, several unsupervised methods for increasingly accurate compound noun splitting are introduced. Bilingual evidence from a parallel German-Polish corpus and co-occurrence counts from the web are used to disambiguate compound noun analyses directly. These collected splits serve as training data for a probabilistic model that abstracts away from the errors made by the direct methods and reaches an f-measure of 95.10%. Furthermore, these methods are evaluated in terms of word alignment quality and extraction accuracy where linguistically accurate methods are found to outperform the corpus-based methods proposed in the literature. A comparison of alignment quality achieved with the best splitting method and the baseline implies that the effort to build supervised splitting methods might result in minimal or no performance gains.

1 Introduction

During the work on the automatic extraction of German compound nouns and their Polish equivalents from a large parallel corpus, we noticed that the splitting of compound nouns has the most beneficial effect on extraction accuracy. A simple splitting method consistently resulted in an improvement of more than 20% for all investigated corpus sizes compared to attempts where no splitting was applied. Encouraged by this result, we investigate whether further improvements can be achieved when more sophisticated splitting methods are employed. We evaluate several unsupervised methods for compound noun splitting using empirical evidence from the corpus and from the web. Using the “one sense per corpus” assumption (Fung 1998) for compound noun constituents as a base, we introduce a probabilistic model of compound nouns that is trained on the data obtained from the direct methods. A second model that allows for exceptions from the previous assumption in the face of strong evidence is proposed. The probabilistic models are shown to outperform the methods they were trained on and reach results only slightly worse than models trained on manually annotated training data.

Our approach to the extraction of bilingual phrase pairs[1] relies on the applica-

[1] For similar approaches to bilingual vocabulary and terminology extraction see for instance Dagan and Church (1998).

tion of the alignment models implemented in GIZA++ (Och and Ney 2003). Polish words that have been aligned with a German compound noun are extracted as equivalents of this noun. We compare the influence of the introduced splitting methods on the quality of alignments and equivalence pairs for a test set of compound nouns that have been manually annotated with their Polish counterparts.

2 Corpus statistics

The corpus we use for our extraction task and for the corpus-based splitting methods is the German-Polish part of the third release of the JRC-Acquis parallel corpus (Steinberger et al. 2006). The JRC-Aqcuis is basically a subset of the *Acquis Communautaire*, the total body of European Union law. The German-Polish language pair compromises 23,322 parallel texts with 1,231,766 alignment links between sentences. In order to reduce the vocabulary size, we deleted links that consisted mainly of foreign language material (i.e. other than German or Polish respectively) in either language. After this deletion the German half of the corpus contains 26,704,419 tokens which correspond to 287,754 types, whereas the Polish half consists of 25,405,924 tokens and 221,014 types. Numeric tokens are ignored. In the German half of the corpus 2,163,620 compound noun tokens were collected which correspond to 142,443 compound noun types. Comparing these numbers with the German corpus statistics, we see that only 8.1% of the tokens are compound nouns, but they account for 50.1% of the overall vocabulary. This is consistent with the findings of other researchers in the field. For instance Baroni et al. (2002) identified 7% of the tokens and 47% of the types of a comparably large newswire corpus to be nominal compounds, whilst Schiller (2005) reports lower percentages (5.5% and 43%) for a newspaper corpus.

3 Working definition of a German compound noun

For our needs we define a German compounded word c (not necessarily a noun) as a string consisting of alphabetical characters and optionally a hyphen, written without spaces that can be split up into a sequence $\mathbf{s}_1^n = s_1 s_2 \ldots s_n$ of n segments. Segments are required to cover the whole string, but are not allowed to overlap. Segments s_1 to s_{n-1} are called *modifier segments*, the last segment s_n is distinguished and is denoted as the *head segment*. We call a sequence of segments a *segmentation*. The set of possible segmentations for a compound noun c is denoted by $\mathrm{Seg}(c)$.

For every segment s there exists at least one corresponding lexeme l. A sequence $\mathbf{l}_1^n = l_1 l_2 \ldots l_n$ of n lexemes where every lexeme l_i corresponds to the segment s_i in the segmentation $\mathbf{s}_1^n$ is called a *decomposition*. Similar to segments, we distinguish between *modifier lexemes* and *head lexemes*. Additionally, we define $\mathrm{Dec}(\mathbf{s})$ as the set of decompositions corresponding to a segmentation $\mathbf{s}$. The set of all possible decompositions of a compound noun c is defined as $\mathrm{Dec}(c) = \bigcup_{\mathbf{s}\in\mathrm{Seg}(c)} \mathrm{Dec}(\mathbf{s})$.

With the help of these definitions the process of compound noun identification can be reduced to a search for strings for which a decomposition into at least two lexemes exists where the head lexeme is a noun. In order to reduce false identifications (for instance *Verbraucher* or *folgende*), the word list produced from the German half of the corpus is filtered. Every word that does not begin with a capital letter or is included in a list of known non-compounded words is discarded.

4 Splitting of compound nouns

4.1 Corpus-based splitting method

Koehn and Knight (2003) propose to consider every capitalized word which can be split into two or more words occurring in the German part of the corpus as a compound word. The inventory of segments is limited to corpus words that are tagged as nouns, verbs, adjectives, adverbs, and negation particles. Compound nouns are allowed to be segments themselves. Originally, no distinctions between modifier and head segments are made. However, head segments are limited to words that have been tagged as nouns. By collecting the frequency $C(s)$ of every segment s in the corpus, the best-scored segmentation $\hat{\mathbf{s}}$ is found as follows:

$$\hat{\mathbf{s}} = \operatorname*{argmax}_{\mathbf{s}_1^n\in\mathrm{Seg}(c)} \sqrt[n]{\prod_{k=1}^{n} C(s_k)} \qquad (8.1)$$

This method will not split a word if its frequency is higher than the geometrical mean of the frequencies of its segments. This makes sense if we have no knowledge of whether a given word is indeed a compound or just a simple word that could be incorrectly split. On the other hand, a number of compound nouns which could be split correctly may remain unsplit. We will refer to this method as CORPUS.

4.2 Lexicon-based splitting method

The source of the segments for the second method is the German-Polish translation lexicon of the POLENG MT system (Jassem 2006) which provides us with inflectional forms of approximately 90,000 non-composed lexemes. The set of head segments simply consists of all inflected forms of non-composed nouns.

Adding correct nominal modifier segments is a more challenging task. According to Fuhrhop (1998) most productive German linking elements are in fact paradigmatic, i.e. the form of a nominal modifier of a compound corresponds to one or more inflected forms of the noun. Usually this is the base form (in most cases), the plural nominative, or the singular genitive. All of these forms are treated as possible modifier segments. Other phenomena at the segmentation border include the addition of *-s* for each base form and the possible omission of a final *e* or combinations thereof. For verbs, adjectives, adverbs, and numerals, the generation of segments is less complicated. Typically it suffices to add the stems and allow for an additional *-e* after verbs with final plosives.

As before we collect frequencies for all segments that can be observed in the corpus, with the remaining segments assigned a frequency of 1. Assigning zero would cause the geometrical mean of the segment frequencies to be zero as well and we would lose the scores implied by other segments in the same segmentation with possibly high frequencies.

$$\hat{\mathbf{s}} = \operatorname*{argmax}_{\mathbf{s}_1^m \in \mathrm{Seg}(c)} \sqrt[m]{\prod_{k=1}^{m} C(s_k)} \tag{8.2}$$

$$\text{where } m = \min\{n : \mathbf{s}_1^n \in \mathrm{Seg}(c) \wedge n > 1\}$$

The scoring function (8.1) is replaced by (8.2). Following Schiller (2005) we prefer the segmentation with the least elements, but not less than two. If there is more than one such segmentation the geometrical mean of word frequencies is used as a back-up. We denote this splitting method as LEX.

This method is used for the identification of compound nouns and the creation of undisambiguated splits and decompositions. About 36.4% of the found compound nouns have only one decomposition, the rest is ambiguous. More than 10 decompositions are possible for about 7%, a small number of compounds nouns (0.1%) have more than 100 decompositions. In most cases only one decomposition makes sense. The large number of mainly spurious decompositions is a negative effect of the simple approach to the generation of segments described above.

5 Disambiguation of decompositions

5.1 Disambiguation by bilingual evidence

Since we are conducting our experiments with splitting methods on compound nouns originating from a parallel corpus, taking advantage of a bilingual dictionary is straightforward for the disambiguation of splitting results. This has been proposed by Koehn and Knight (2003), who employ an automatically extracted dictionary in combination with their corpus-based splitting method for the reduction of structural ambiguities.

Contrary to Koehn and Knight (2003), we search for the single best decomposition common for all tokens of one compound noun. For this purpose the translational evidence that is available in all sentence pairs in which the compound noun appears is taken into account simultaneously. The best decompositions are those for which evidence for the greatest number of lexemes is found most frequently. If no evidence has been found or if for several lexemes the same number of translations has been identified, the ambiguities are preserved. This disambiguation method is denoted as +DIC.

For this method the choice of an appropriate bilingual dictionary is crucial. Experiments with different dictionaries, hand-crafted and automatically extracted from GIZA++ translation tables, showed that the manually composed POLENG dictionary performs best. The noise in automatically produced dictionaries has a negative impact that persists even when thresholds are used.

5.2 Disambiguation by web-counts

The application of web statistics to the interpretation and bracketing of English compound nouns has been described by Lapata and Keller (2004). We test a similar approach to the disambiguation of splitting options of German compound nouns. Hit counts retrieved from GOOGLE for appropriately constructed queries serve as information about co-occurrences of compound noun segments and corresponding lexemes. The decomposition that receives the highest number of hits and its underlying segmentation are marked as accepted. The methods rely on two types of queries:

- Queries consist of selected forms of the lexemes belonging to a decomposition. For nouns the base form is used, for verbs and adjectives we use inflected forms to avoid confusion with homonymous nouns. This disambiguation method is marked as +WWW$_1$.

- Queries include all search keys from the previous method. Apart from that, the unsplit compound noun is added. This method is named +WWW$_2$.

Both types of search requests can be cascaded in cases where the queries extended with the unsplit compound noun return less than five hits for all decompositions. We then drop the compound noun from the query and repeat the search. This results in the method named +WWW$_3$.

The common weakness of all these approaches is their inability to disambiguate homonymous lexemes for which identical queries are generated. Also differences in the capitalization of the first letter, an important clue for the distinction of nouns from other words, cannot be captured this way.

5.3 Combined disambiguation

All of the disambiguation methods introduced preserve ambiguities if there was not enough evidence to choose a single best lexeme for a segment. We can assume that the described disambiguation methods fail for different compound nouns. For instance, homonymous lexemes can be resolved easily by dictionary look-up provided appropriate entries are available; data sparseness is hardly a problem for the web-based methods, but they cannot deal with homonyms. Therefore, a combined approach should improve the general results. We cascade both methods in the following way:

- The first disambiguation method is applied to the analysis produced by a chosen splitting method.
- Only the best scored results are kept. If there are no ambiguities, the single best result has been found.
- If the remaining results are still ambiguous, the second method is applied.

According to our naming convention, we label the lexicon-based splitting method as LEX+DIC+WWW3 where the dictionary-based disambiguation method +DIC is applied before the web-based method +WWW3.

6 A probabilistic splitting method

In this section a probabilistic model of compound nouns that uses the splitting knowledge acquired by the direct approaches as training data will be described. In a first step the compound nouns collected are analyzed using the method LEX+DIC+WWW3 and only the best scored results are stored. Table 1 shows a sample of the collected data. $C_{\mathrm{M}}(s)$ denotes the number of times a modifier segment s occurred in all segmentations and $C_{\mathrm{M}}(s,l)$ counts how often a lexeme l was assigned to this segment. Fractional counts are added if lexical ambiguities could not be fully resolved. Obviously $C_{\mathrm{M}}(s) = \sum_l C_{\mathrm{M}}(s,l)$. The counts for heads, $C_{\mathrm{H}}(s)$ and $C_{\mathrm{H}}(s,l)$, are collected analogously.

For the modifier segment *rechts* the lexeme *Recht_N* (*law*) was assigned in 87.7% of the splits, but *Rechte_N* (*right hand* or *person with right-wing views*) was assigned in more than 7.2% of the splits which is still more than we would expect in a corpus

Table 1. Counts for *recht* and *steuer* as modifiers and heads.

Segment	$C_{\mathrm{M}}(s)$	Lexeme	$C_{\mathrm{M}}(s,l)$	Segment	$C_{\mathrm{H}}(s)$	Lexeme	$C_{\mathrm{H}}(s,l)$
rechts	814	Recht_N	731.7	rechts	203	Recht_N	202.0
		Rechte_N	58.8			Rechte_N	1.0
		rechts_adv	23.5	steuer	106	Steuer_N	99.5
steuer	730	Steuer_N	596.5			Steuer_N2	6.5
		Steuer_N2	103.0				
		steuern_V	30.5				

of law-related texts. A manual check reveals that all assignments of *Rechte_N* to *rechts* are indeed incorrect, and similarly for *rechts_adv* (*on the right side*) where only *Rechtslenker* (*right-hand drive vehicle*) was correctly analysed. These errors are due to incorrectly generated segments, as for *Rechte_N*, or true ambiguities, as in the case of *rechts_adv*. In 813 out of 814 cases *Recht_N* would have been the correct choice for *rechts*.

6.1 One sense per corpus

This example suggests that the "One sense per corpus" hypothesis introduced by (Fung 1998) in the context of bilingual vocabulary extraction can also be applied to the disambiguation of compound noun constituents. Choosing the most probable lexeme for a segment complies with this hypothesis.

The probability of a modifier segment $Pr_{\mathrm{M}}(s)$ and the probability of a lexeme corresponding to a modifier segment $Pr_{\mathrm{M}}(l|s)$ are calculated as follows

$$Pr_{\mathrm{M}}(s) = \frac{C_{\mathrm{M}}(s)}{\sum_{s'} C_{\mathrm{M}}(s')}, \quad Pr_{\mathrm{M}}(l|s) = \frac{C_{\mathrm{M}}(s,l)}{\sum_{l'} C_{\mathrm{M}}(s,l')} \tag{8.3}$$

using simple Maximum Likelihood Estimates. Again, the probabilities for head segments and corresponding lexemes, $Pr_{\mathrm{H}}(s)$ and $Pr_{\mathrm{H}}(l|s)$, are calculated similarly. Smoothing methods are not applied. We can now express the probability of a single segmentation $\mathbf{s}_1^n$ by

$$Pr(\mathbf{s}_1^n) = \prod_{k=1}^{n-1} Pr_{\mathrm{M}}(s_k) Pr_{\mathrm{H}}(s_n). \tag{8.4}$$

The probabilistic equivalent of the scoring function (8.2) which allows us to choose the best segmentation can be stated as

$$\hat{\mathbf{s}} = \operatorname*{argmax}_{\mathbf{s} \in \mathrm{Seg}(c)} Pr(\mathbf{s}). \tag{8.5}$$

Analogously as for segments, we define the probability of a decomposition $\mathbf{l}_1^n$ given a segmentation $\mathbf{s}_1^n$ as

$$Pr(\mathbf{l}_1^n|\mathbf{s}_1^n) = \prod_{k=1}^{n-1} Pr_{\mathrm{M}}(l_k|s_k) Pr_{\mathrm{H}}(l_n|s_n). \tag{8.6}$$

Finally for the probability of a compound c corresponding to a sequence of lexemes $\mathbf{l}_1^n$ we have

$$Pr(\mathbf{l}_1^n|c) = \sum_{\mathbf{s}_1^n \in \mathrm{Seg}(c)} Pr(\mathbf{s}_1^n) Pr(\mathbf{l}_1^n|\mathbf{s}_1^n). \tag{8.7}$$

Since it is possible (although unlikely) that one decomposition is generated by more than one segmentation of the same compound noun, we sum over all of the found segmentations. Equation (8.7) offers us the means to search for the best decomposition $\hat{\mathbf{l}}$ of a given compound noun without the need of consulting external data. This is done in an analogy as for the best segmentation in equation (8.5) by

$$\hat{\mathbf{l}} = \operatorname*{argmax}_{\mathbf{l} \in \mathrm{Dec}(c)} Pr(\mathbf{l}|c). \tag{8.8}$$

6.2 Weakening independence

As can be seen in equations (8.4) and (8.6), segments on different positions in a segmentation as well as the corresponding lexemes in a decomposition are assumed to be independent. This means that for a given modifier segment the same lexeme is chosen regardless of the choice made for the other segments and *vice versa*. Although this is consistent with the aforementioned “one sense per corpus” assumption for compound noun constituents, we would still prefer to be able to account for exceptions in cases when we have strong evidence for them. One example is *Steuergerät*, where the decomposition *steuern_V Gerät_N* (*steering device*) makes more sense than *Steuer_N Gerät_N* (*tax device*).

In order to weaken independence, we introduce another probability distribution assuming that head lexemes have preferences concerning the semantics of their modifiers, a tendency that has been described by Langer (1998). Since no bracketing methods are used, the linear order of lexemes in a decomposition is the only structural information available, and we adopt the oversimplifying assumption that lexemes restrict the semantics of the lexeme that directly precedes them. This is in fact equivalent to expecting a left-branching binary structure for all compound nouns which should be correct for approximately 90%.[2]

Table 2. Example results for $c = \textit{Steuergerät}$ without (*) and with (**) semantic context

l_1	l_2	$Pr(l_1 l_2 \vert c)$ (*)	l_1^{sem}	$Pr(l_1^{\text{sem}} \vert l_2)$	$Pr(l_1 l_2 \vert c)$ (**)
Steuer_N	*Gerät_N*	**1.9093e-06**	POSSESSION	0.0012	2.3699e-09
Steuer_N2	*Gerät_N*	3.2736e-07	ARTIFACT	0.1207	3.9523e-08
steuern_V	*Gerät_N*	9.6950e-08	ACT	0.5135	**4.9782e-08**

The semantic information is obtained from the POLENG lexicon, which is organized in an ontology that features more than 130 concepts. In order to avoid data sparseness, the concepts are mapped to a set of 17 concepts from the first three levels of the ontology tree. The concept assigned to a lexeme l is denoted by l^{sem}. Proceeding on the mentioned assumptions, we express the semantic probability of a decomposition $\mathbf{l}_1^n$ by

$$Pr_{\text{SEM}}(\mathbf{l}_1^n) = \prod_{k=2}^{n} Pr(l_{k-1}^{\text{sem}} | l_k) \tag{8.9}$$

and replace equation (8.7) with

$$Pr(\mathbf{l}_1^n | c) = Pr_{\text{SEM}}(\mathbf{l}_1^n) \sum_{\mathbf{s}_1^n \in \text{Seg}(c)} Pr(\mathbf{s}_1^n) Pr(\mathbf{l}_1^n | \mathbf{s}_1^n). \tag{8.10}$$

As before, the empirically disambiguated decompositions are used as training data. Even a frequent lexeme like *Gerät_N* does not co-occur with all semantic concepts. Although we do not use the splitting method on compound nouns from outside the corpus, using MLE for the estimation of $Pr(l_{i-1}^{\text{sem}} | l_i)$ might introduce a number of zero probabilities in equation (8.9), due to data sparseness, and worse, due to errors produced by the empirical disambiguation methods. In order to avoid this influence of Pr_{SEM} on the scoring function (8.10), we apply smoothed counts computed with the help of the Simple Good-Turing (SGT) method (Gale 1994), calculating

$$Pr(l_{i-1}^{\text{sem}} | l_i) = \frac{C_{\text{SGT}}(l_{i-1}^{\text{sem}}, l_i)}{\sum_{l'_{i-1}} C(l'_{i-1} l_i)}. \tag{8.11}$$

Table 2 illustrates the differences between the two probabilistic models defined by equations (8.7) and (8.10) for our previous example *Steuergerät*. As said before, the decomposition that maximizes equation (8.7) is *Steuer_N Gerät_N*. Introducing contextual knowledge by equation (8.10) allows us to identify *steuern_V Gerät_N*

[2] 72% of the examined compounds are of length 2 where the branching structure is irrelevant. For the rest we can assume that about two-thirds are left-branching. This has been shown to be true for complex English compound nouns by Lauer (1995), and we assume a similar distribution for German.

as a better decomposition. Since the semantic concept POSSESSION was unseen for any lexeme preceding *Gerät_N* and the probability for the concept ACT appearing before *Gerät_N* is very high, we have strong evidence for an exception from the "one sense per corpus" assumption. In most cases, however, the influence of the additional semantic context is only marginal.

7 Evaluation

In this section we present the evaluation of the proposed splitting methods under two different aspects: the linguistic correctness of the analyses and the influence of the splitting methods on the quality of bilingual alignment. The alignments are computed with GIZA++ (Och and Ney 2003). We use refined alignments produced from two alignment models trained in both directions as has been proposed by Och and Ney (2003). The input to GIZA++ has been lemmatized in order to reduce the size of the vocabulary.

7.1 Compound noun splitting

The performance of the described methods is evaluated on a test set of $\mathrm{N} = 1000$ compound nouns that have been manually annotated with correct segmentations and decompositions. For segmentations accuracy is the only measure used. Correct segmentations (cr) are those for which all splitting points are identical with the splitting points in the manually created split. Accuracy is then calculated as $\mathrm{A} = \mathrm{cr}/\mathrm{N}$.

In order to compare our results with a supervised splitting method, we use the same evaluation scheme for decompositions as has been proposed by Schiller (2005). Only "best scored" decompositions are taken into account. If there is no scoring method for decompositions (see CORPUS or LEX), all decompositions of the best segmentation are considered as results.

Among the set of results returned for one compound noun, the analyses which are identical with the manually disambiguated decomposition are true positives (tp) — in our case there is at most one for each compound noun. All other analyses that do not match the manual choice count as false positives (fp). Additionally, a false negative (fn) is counted if the manual analysis is not among the results. Given the above values for all compound nouns in the test set, we calculate the overall precision (P), recall (R), and f-measure (F) in the standard way, where $\mathrm{P} = \mathrm{tp}/(\mathrm{tp}+\mathrm{fp})$, $\mathrm{R} = \mathrm{tp}/(\mathrm{tp}+\mathrm{fn})$, and $\mathrm{F} = (2\cdot\mathrm{P}\cdot\mathrm{R})/(\mathrm{P}+\mathrm{R})$.

Table 3 summarizes the results for the proposed splitting and disambiguation methods. As for segmentations, the task of correctly choosing all splitting points in a compound noun, all approaches based on the LEX method perform reasonably well.

Table 3. Percentages for splitting accuracy, precision, recall, and f-measure

Splitting method	Segmentations	Decompositions		
	Accuracy (%)	P (%)	R (%)	F (%)
LEX (baseline)	98.70	67.23	97.74	79.66
LEX+DIC	98.70	78.59	96.93	86.80
LEX+WWW$_1$	98.70	89.10	90.70	89.89
LEX+WWW$_2$	99.00	86.58	93.92	90.10
LEX+WWW$_3$	99.00	90.67	93.21	91.92
LEX+DIC+WWW$_3$	99.00	92.38	94.31	93.34
LEX+PROB	99.30	94.35	94.73	94.54
LEX+PROB+SEM	**99.30**	**95.05**	**95.15**	**95.10**
CORPUS	72.15	40.70	62.39	49.26
CORPUS+DIC	72.15	44.93	59.62	51.24

For the web-based methods it seems reasonable to cascade disambiguation methods in such a way that the method with higher precision which is simultaneously more prone to data sparseness precedes the less precise but more robust method. This is also true for the combination of dictionary look-up and web-counts which results in the best empirical splitting method LEX+DIC+WWW$_3$ with an f-measure of 93.34%.

Our claims from section 6.1 concerning the validity of the "one sense per corpus" hypothesis for compound noun elements seem to be confirmed by the results achieved by the probabilistic models. Abstracting from the data obtained from the unsupervised methods leads to an improved precision for LEX+PROB compared to the best empirical method. Method LEX+PROB+SEM additionally incorporates knowledge about exceptions from the above assumption and performs best among all investigated unsupervised approaches to compound noun splitting and analysis.

As for supervised methods, Schiller (2005) uses weighted finite state automata trained on two large sets of manually split and disambiguated compound nouns from newspaper texts. For an in-domain test set, f-measures between 98.32% and 98.38% are reported. Our best result for decompositions, an f-measure of 95.10%, is not as high but still acceptable.

The results for the CORPUS and CORPUS+DIC methods are given for reasons of completeness. Since the aim of these methods is to establish one-to-one correspondences between bilingual equivalents rather than to provide linguistically correct analyses, they cannot be compared directly to our splitting methods. These results are nevertheless significant when we compare the alignment quality for different splitting methods in the next section.

Table 4. Percentages for Alignment Error Rate and Extraction Accuracy for chosen splitting methods

Splitting method	100K sentences		500K sentences		1.2M sentences	
	AER	EA	AER	EA	AER	EA
No splitting	56.38	21.5	48.19	27.9	—	—
CORPUS	31.87	37.9	25.51	45.6	—	—
CORPUS+DIC	31.61	38.2	25.49	45.7	—	—
LEX	26.64	43.8	20.91	51.2	17.58	56.2
LEX+PROB+SEM	25.99	44.8	20.56	51.4	17.57	56.1

7.2 Word alignment and extraction quality

We now turn to the original question how compound noun splitting quality affects the alignment quality for compound nouns. For the alignment and extraction task we annotated 1000 German compound nouns with their Polish equivalents. The compound nouns were randomly chosen from the identified compound nouns types, after which one token was selected for each type. There are no repetitions in the test set and it is distinct from the test set used in the previous section. In compliance with Och and Ney (2003), we distinguish between sure alignments (S) and possible alignments (P, where $S \subseteq P$) that additionally describe ambiguities, such as function words that are missing in the other language. We do not annotate the whole sentence that contains the compound noun, but only the alignment points associated to the given compound noun itself.

Och and Ney propose the following measures to calculate precision, recall, and alignment error rate (AER) for the sets S and P from the test set and an alignment A obtained from the word alignment process:

$$\mathrm{P} = \frac{|A \cap P|}{|A|}, \quad \mathrm{R} = \frac{|A \cap S|}{|S|}, \quad \mathrm{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}. \tag{8.12}$$

We define an additional measure for the quality of the extraction of the German-Polish translation pairs. Extraction accuracy (EA) is the percentage of compound nouns from the test set that were correctly and completely aligned with their sure Polish equivalent. These are raw numbers calculated from the original alignment data before the application of additional filtering and reconstruction techniques.

Table 4 presents the results for chosen splitting methods and various corpus sizes.[3] The general positive impact of compound noun splitting is obvious for all

[3] For technical reasons for the complete corpus results are evaluated for the baseline and the best splitting method only. The computation for five different splitting methods with two directional models each would have been too time consuming.

corpus sizes and was to be expected. What is more surprising is the significant superiority of the lexicon-based methods compared to the corpus-based splitting approaches. It can be clearly seen that one-to-one correspondence is less beneficial than linguistically motivated splitting. The improvement introduced by the linguistically more correct method LEX+PROB+SEM over the baseline method LEX is more significant for the smaller sets of training sentences and decreases with increasing corpus size.

8 Conclusions

We have shown two things. Firstly, unsupervised methods for compound noun splitting can reach results close to the performance of methods trained on manually disambiguated data. However, it would be interesting to see how the probabilistic models behave when the size of the training data varies. Approximately 140,000 compound nouns were disambiguated using the described empirical methods. Since no human interaction is required, it is no problem at all to increase the amount of training data, for instance with the release of a future, larger version of the JRC-Acquis.

Secondly, splitting methods that aim for full linguistic analyses of compound nouns achieve better results in alignment quality than methods that try to establish one-to-one correspondences. The reasons for this may be due to better coverage of segments, the consistency of splits for all tokens of the same compound noun, and a reduced vocabulary since no compound noun remains unsplit. For smaller corpora the linguistically most accurate splitting method achieves better results than our baseline method, but the effect diminish with increasing corpus size. For large corpora the performance jump of 15% for splitting quality is not reflected in alignment quality or extraction accuracy. The statistical alignment models seem to be able to cope with minor inaccuracies concerning compound noun splitting. This implies that unsupervised methods are sufficiently accurate and no improvement could be achieved by employing models trained on better and therefore more costly data.

References

Baroni, Marco, Johannes Matiasek, and Harald Trost (2002). Predicting the Components of German Nominal Compounds. In *Proceedings of the 15th European Conference on Artificial Intelligence, ECAI'2002*, 470–474.

Dagan, Ido and Ken Church (1998). Termight: Coordinating Humans and Machines in Bilingual Terminology Acquisition. *Machine Translation* 12(1-2):89–107.

Fuhrhop, Nanna (1998). *Grenzfälle Morphologischer Einheiten*. Tübingen, Germany: Stauffenburg.

Fung, Pascale (1998). A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-Parallel Corpora. In *AMTA '98: Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup*, 1–17.

Gale, William (1994). Good-Turing Smoothing Without Tears. Statistics Research Reports from AT&T Laboratories 94.5, AT&T Bell Laboratories.

Jassem, Krzysztof (2006). *Przetwarzanie tekstów polskich w systemie tłumaczenia automatycznego POLENG*. Poznań, Poland: Wydawnictwo Naukowe UAM.

Koehn, Philipp and Kevin Knight (2003). Empirical Methods for Compound Splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL).*, 347–354.

Langer, Stefan (1998). Zur Morphologie und Semantik von Nominalkomposita. In *Tagungsband der 4. Konferenz zur Verarbeitung natürlicher Sprache*, 83–97.

Lapata, Mirella and Frank Keller (2004). The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, 121–128.

Lauer, Mark (1995). Corpus Statistics Meet the Noun Compound: some Empirical Results. In *Proceedings of the 33rd annual meeting of the Association for Computational Linguistics*, 47–54.

Och, Franz Josef and Hermann Ney (2003). A systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29(1):19–51.

Schiller, Anne (2005). German Compound Analysis with *wfsc*. In *Finite-State Methods and Natural Language Processing, 5th International Workshop, FSMNLP 2005*, 239–246.

Steinberger, Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Daniel Varga (2006). The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages. *CoRR* abs/cs/0609058, informal publication.

A taxonomy of task-related dialogue actions: the cases of tutorial and collaborative planning dialogue

Magdalena Wolska and Mark Buckley

Abstract. In this paper we present a taxonomy of dialogue moves which describe the actions performed by participants in task-oriented dialogue. Our work is motivated by the need for a categorisation of such actions in order to develop computational models for tutorial dialogue. As such, we build on both existing work on dialogue move categorisation for the tutorial genre as well as general dialogue act taxonomies. Our taxonomy has been prepared by analysing a corpus of tutorial dialogues in mathematics. We present a top level taxonomy for task-oriented dialogue and show how it can be instantiated for data from both tutoring and, as an example, collaborative planning. We also detail an annotation experiment in which we apply the instantiated taxonomies to validation data, present example annotations, and discuss idiosyncrasies in the data which influence the decisions in the dialogue move classification.

1 Introduction

The field of Intelligent Tutoring Systems has seen recent developments moving towards adding natural language capabilities to computer-based tutoring. However, to be able to interact with a student through the medium of natural language dialogue, the system must have a model of how such tutorial dialogues can progress and what utterances are licenced. In order to develop such a model of dialogue, we need to understand and describe the “actions” performed with words, i.e. speech acts (Austin 1955) or *dialogue moves*. This involves identifying and categorising the functions that utterances may have in dialogue and their relationships to each other. Researchers in conversation and dialogue analysis have proposed various general categorisations of dialogue moves. DIT++ (Bunt 2000) is an example of a comprehensive multidimensional taxonomy of dialogue acts for information dialogues based on DAMSL (Allen and Core 1997), a general-purpose extensible taxonomy proposed as a standard for dialogue annotation.

The DAMSL taxonomy characterises utterances along four dimensions, which correspond to four levels of functions utterances may have. The forward looking function describes the utterance’s effect on the following interaction and the backward looking function its relation to previous dialogue. The communicative status describes the comprehensibility or interpretability of the utterance, and the information level characterises the content of the utterance. Tsovaltzi and Karagjosova (2004) proposed an extension of the DAMSL classification based on an analysis of tutorial dialogue corpora. Their taxonomy adds a **Task** dimension which concentrates on tutor actions in the dialogue.[1] Building on this work, we first observe that a task level classification of dialogue actions covering general task level functions can be specified. Second, we show that this classification can be instantiated depending on the domain and genre of the dialogue, in particular for tutorial dialogue.

The classification we present includes (i) modifications of the DAMSL categorisation motivated by tutorial dialogue and (ii) accounts for student's actions, however (iii) unlike Tsovaltzi and Karagjosova (2004) we do not use a separate task dimension but rather retain the DAMSL structure. Furthermore, we also show that our taxonomy can be instantiated for an another dialogue genre, namely collaborative problem solving. We validated our taxonomy for tutorial dialogue in a small-scale annotation experiment, whose results we present. The work presented here continues our previous work (Buckley and Wolska 2008b) by abstracting and restructuring the taxonomy. Both this and the work in the current paper are additionally motivated by our analysis of the structures used by students and tutors in tutorial dialogues (Buckley and Wolska 2008a).

An overarching goal of this work is to support parametrisable dialogue expertise, and so dialogue modelling. Based on the top level taxonomy general dialogue rules can be defined. Like the categories in the taxonomy, these can be instantiated (parametrised) for a given genre or domain, for instance in terms of the conditions under which they can be performed, the obligations they impose on hearers, or the effects they have on speakers' mental states. This paper is organised as follows: In Section 2 we introduce our data and taxonomy development methodology. In Section 3 we present the taxonomy. The results of the annotation experiment and their discussion are presented in Section 4. Section 5 concludes the paper.

2 Materials and methods

Our work is based on an analysis of data from two dialogue genres: tutoring and problem-solving. We use the data to (i) verify the general dialogue dimensions of the DAMSL taxonomy in the context of tutorial dialogues, and (ii) extend the taxonomy by developing the task dimension for the two genres. While the specific task-level moves for tutoring and problem-solving are instantiated for mathematics tutoring and collaborative planning, respectively, our aim is to maintain generality that would make it possible to model task-oriented dialogues in general, in other words other tutoring domains as well as other problem solving domains. We now briefly introduce the corpora and outline our methodology.

[1] This classification has not been, to our knowledge, quantitatively evaluated.

S1	$R \circ S := \{(x,y) \mid \exists z(z \in M \wedge (x,z) \in R \wedge (z,y) \in S)\}$
T1	That's right!
S2	now i want the inverse of that
T2	yes?
S3	$(R \circ S)^{-1}$
T3	= ?
S4	How will the system answer?
T4	What's the question?

Figure 1. Example dialogue from Corpus-II

2.1 Data

The general taxonomy as well as its two instantiations have been developed based on three sets of dialogue data. The tutoring taxonomy was developed based on data from mathematical proof tutoring and validated for scalability on data from the same genre (mathematics tutoring) however involving a computational task (differentiation). The second genre-specific instance of the taxonomy was developed based on dialogues on problem solving, in particular, collaborative planning. The instantiation for a different dialogue genre serves to show that the top-level taxonomy provides enough generalisation to be applicable to new domains. We now briefly introduce each corpus in turn.

Proof tutoring data Our analysis of proof tutoring is based on excerpts from two corpora of tutorial dialogues in the domain of mathematical theorem proving collected in Wizard-of-Oz experiments in the DIALOG project[2]. The domain of mathematics in the first (Corpus-I) Wolska et al. (2004) and second corpus (Corpus-II) Benzmüller et al. (2006) are naive set theory and binary relations respectively. In both experiments the dialogues were conducted in German using the keyboard and a graphical user interface. Corpus-I contains dialogues conducted in three experimental conditions: minimal feedback, didactic or socratic tutoring strategy. The verbosity of the *minimal feedback* tutors was limited, while in both other conditions as well as in the second experiment, the subjects and the tutors were unconstrained in terms of the linguistic realisation of their turns. To illustrate the type of data in the theorem proving corpora we give an excerpt in Figure 1, which contains a discussion of an inverse operation. Corpus-I comprises 775 turns (332 student and 443 tutor turns, respectively), Corpus-II has 1917 turns (937 student and 980 tutor turns). More details on the proof tutoring corpora and the experiments can be found in Benzmüller et al. (2003); Benzmüller et al. (2006).

[2] `http://www.ags.uni-sb.de/~dialog/`

Differentiation tutoring data The analysis of dialogues in a related tutoring domain is based on excerpts of human-human dialogues collected within the LeActiveMath project[3]. The LeActiveMath corpus consists of 33 transcripts of tutoring sessions on differentiation, conducted via a chat interface. The tutors were five experienced mathematics instructors and the subjects were first-year mathematics or science undergraduate students (28 subject, of whom 5 participated twice). Mathematical expressions were entered using a formula editor, and text and formulas could be interleaved. The corpus contains 1650 utterances. Further details on this data can be found in Callaway and Moore (2007); Porayska-Pomsta et al. (2008).

Problem solving data Our third data set was taken from a corpus of problem solving dialogues collected within the TRAINS project[4]. The dialogues involve two human participants: one participant plays a role of a user who has a certain task to accomplish, and the other plays the role of the system by acting as a planning assistant in a transportation logistics task. The corpus contains 98 dialogues involving 20 different tasks (5900 speaker turns). The experimental setup and the corpus have been presented in detail by Heeman and Allen (1995).

2.2 Methodology

In developing the general taxonomy and its instantiations for the tutoring and the problem-solving domains we pursued the following methodology:

First, in order to build the initial general and tutoring-specific taxonomy, we analysed 18 dialogues from Corpus-I (the development set, consisting of 299 utterances). The purpose of this analysis was to (i) verify the general suitability of the DAMSL scheme in the tutoring domain[5], (ii) identify features of dialogues moves relevant in tutoring that were not present in the original taxonomy (see the discussion in Section 4), (iii) identify an initial set of general and tutorial dialogue specific task-level moves. We descriptively defined the tutoring move types and wrote draft annotation guidelines. We applied the initial tutoring taxonomy to 4 dialogues (108 utterances) taken from both proof corpora in an annotation task performed independently by both authors of this paper (a preliminary test set), after which we both extended the taxonomy and refined the existing category definitions.

Second, in order to test the coverage of the final tutoring taxonomy, we randomly selected a 64-utterance subset of both proof corpora[6] (the validation set) which we annotated independently. The results of this annotation are presented in Section 4.

[3] `http://www.activemath.org`

[4] `http://www.cs.rochester.edu/research/cisd/projects/trains/`

[5] We expected this to be suitable because it is a taxonomy with general applicability.

[6] We avoided utterances consisting of formulas only.

To ensure scalability of the taxonomy, we applied the tutoring instantiation of the taxonomy to 74 utterances from the LeActiveMath corpus (within-genre validation). We discuss this annotation in Section 3.2. Finally, we analysed 2 dialogues (67 utterances) from the TRAINS corpus to build the initial instantiation of the task-level taxonomy for the collaborative planning domain. In order to validate this taxonomy instance, we applied it to 63 unseen utterances from the same corpus (cross-genre validation). The taxonomy resulting from this study is presented in Section 3.3.

3 A dialogue move taxonomy

Our goal in the analysis of the development set and preliminary test set of the corpus was to determine a categorisation of the actions that can be performed by students and tutors in tutorial dialogues, keeping in mind that the actions instantiate a top level taxonomy of general task level actions. The taxonomy which we have created from this categorisation contains the dialogue moves which realise these actions. The utterances performed by students and tutors realise actions which may or may not address or have an effect on the current task. We therefore speak of the task-level function of an utterance in addition to a general dialogue level function which all utterances have. The split between task-level and general dialogue level function which exists in DAMSL is preserved in our taxonomy. We first present the taxonomy at an abstract level and then show that it can be instantiated to two different genres: tutorial dialogue and problem solving dialogue.

3.1 The abstract taxonomy

At the general dialogue level we follow the DAMSL taxonomy and categorise the functions of utterances according to their relationship with the previous dialogue and their effect on the dialogue to follow. For these functions we use a forward dimension and a backward dimension, respectively. In general, we try to accommodate the DAMSL categories in order to build as much as possible on existing generally accepted work on dialogue moves. The forward dimension captures utterances which are either assertions, requests or commands. The backward dimension captures utterances which agree or disagree with previous utterances, address questions, signal the understanding status of previous utterances, or stand in some information relation to previous utterances. The main differences between DAMSL and our categorisation within these two dimensions are the following: (i) we combine DAMSL's Assert and Re-assert in a single category Assert which may be optionally marked as repeating information, (ii) we combine DAMSL's Action-directive and Info-request in a higher-level category of Requests, (iii) in place of DAMSL's Answer, we introduce a more general **Address** category in the backward dimension with subcategories **Answer**, **Deflect**, and **Neutral**, where **Deflect** accounts for avoiding answering and **Neutral** refers to those utterances which simply address a previous information request without answering it or a previous action directive without acceding to it. The remaining DAMSL categories are left unchanged.

DAMSL uses an information-level dimension to characterise the content of the utterance, which can be related to the task, task management or communication management. Communication management refers to conventional phrases like salutations or references to the communication channel, and is not further specified here. Utterances which contribute to the task at hand are of type **Task**, those which contribute to the task solving process are **Task Management**. It is these two categories which we will further specify in order to enrich the DAMSL taxonomy to cover tutorial dialogue. Utterances in the task category have the function of altering the state of the task solution, for instance by performing a step in the solution, or talking about the task solution without altering it, for instance making statements about previously performed steps. We divide the task related actions in those which address the task directly and those which address the solution construction process, and capture these in the task and task management categories respectively.

The notion of what is a contribution in the given dialogue genre, what task-level functions are relevant and what types of task-level contributions there are is determined by both the type of dialogue and participants' goals. In tutoring dialogues, task-level contributions refer to the solution to a posed problem. Therefore we have identified the *solution step* as the building block of tasks in tutoring in formal domains. The participants' goals depend on their role as either student or tutor. The student's goal is to solve a problem (e.g. prove or compute), while the tutor's goal is to teach (e.g. guide towards a solution by giving hints). In other genres, the participants' roles, their goals, and so the definition of a contribution, are different and depend on the dialogue purpose. For instance in collaborative problem solving the participants' roles are those of planning agents and their goals are to construct a plan for the problem at hand. The main dialogue contribution is thus a step in a plan. In information seeking dialogue (such as time-table enquiries), in which the participants' roles are those of information "seeker" and "giver" and the goals are to find out and provide information, respectively. The contributions focus on enquiry components.

In Table 1 we present the abstract top level taxonomy with explanations of the labels and examples. The **Task** category has three subcategories: **Contribute domain content** covers utterances which bring new domain content into the dialogue, such as new or continued solution steps. **Address domain content** labels utterances which talk about previously introduced solution steps, for instance evaluations. **Request** covers a number of task utterances in which speakers ask for or about concepts in the domain. The **Task management** category contains dialogue moves which start, finish, restart or give up the task, as well as moves which refer to the status of the

Table 1. The full taxonomy. Each type is given along with an explanation and an example

Label	Explanation	Example
Forward Dimension		
Assert	Makes a claim about the world	“It holds that P”
Request	Introduces an obligation to answer	
Action-directive	The obligation is that an action is performed	“Please show the following ”
Info-request	Request for a piece of information	“What is the definition of...?”
Open-option	Suggestion of future action without obligation	“You could do . . . ”
Backward Dimension		
Agreement	Acceptance or rejection of propositions	
Accept	Accepts a proposal	“Ok, that’s right”
Reject	Rejects a proposal	“that’s incorrect”
Address	Responses to requests	
Answer	Answers a previously posed info-request	“yes” or “no”
Deflect	Shows inability or unwillingness to answer	“I can’t answer that”
Neutral	Addresses without answering or deflecting	“Why do you ask?”
Information relation	Relation to an antecedent utterance	
Understanding related	Refers to problems understanding the speaker	
request clarification	Asks to clarify a previous utterance	“what do you mean by X?”
request rephrase	Asks for a repeat/rephrase of an utterance	“could you repeat that?”
signal non-understanding	Catch-all for understanding problems	“pardon?”
Information-level Dimension		
Communication management	Maintaining communication/contact	”Hello!”
Task	Performing the task	
Contribute domain content	Refers to content for the current solution	“let $x \in A$”
Address domain content	Discuss a performed step	“Good idea!”
Request	Ask for help or information	“What’s the next step?”
Task-management	Addressing the task-solving process	
Start task	Starts the solution construction process	“please prove $P = Q$”
Finish task	Indicates end of solution construction	“I’m done”, “q.e.d”
Restart task	Indicates solution being started again	“start again”
Give-up task	Abandons the current solution attempt	“I give up”
Task solution status	References to solution progress	“Your solution’s not finished”
Check solution adoption	Check is solution is acceptable	“So shall we do that?”
Other		

task, for instance whether it is complete or not.

3.2 Instantiating the taxonomy for tutorial dialogue

Table 2a presents the instantiation of the task category of the taxonomy for tutorial dialogue. Domain content in this genre is realised by solution steps and strategies. New steps can be performed and existing steps can be augmented, for instance with missing parameters. In addition, the category **Provide domain content** is used for domain content which does not perform a solution step, such as references to objects which are part of the general state of affairs. After having been performed steps can be addressed and discussed. Typical in tutorial dialogue are evaluations of steps, for instance as correct or incorrect. Further information about a step can be elicited or a hint can be given. There are many types of requests in tutorial dialogue; our taxonomy lists those which occur in the data, but this list is not necessarily exhaustive.

Table 2. Instantiations of the task dimension for (a) tutorial dialogue and (b) collaborative planning dialogue

(a)

Label
Contribute domain content
Perform step
New solution step
Solution step augmentation
Provide domain content
State of Affairs
State strategy
State strategy
State future step
Address domain content
Evaluate step
Correct
Incorrect
Elicit further step information
Hint
Request
Req explanation
Concept def
Symbol
Req worked example
Req domain content
Step
Solution strategy
Step augmentation
Reformulation

(b)

Label
Contribute domain content
Perform step
Propose plan or plan step
Augment plan step
Table a plan step
Provide domain content
State of affairs
State strategy
State new subgoal to be planned
Address domain content
Evaluate plan proposal
Accept/reject proposal
Verify adoption
Request
Req information
State of affairs
Operator
Object (parameter)
Req next step

Both students and tutors can request explanations of domain concepts or worked examples. Students can also request that the tutor supply steps or strategies.

3.3 Instantiating the taxonomy for collaborative planning dialogue

In Table 2b we present the instantiation of the task category for the genre of collaborative planning dialogue. As we introduced above, these categories are the result of an annotation of 2 dialogues from the TRAINS corpus. Since the goal of the dialogue is to construct a plan, we have defined the solution step in this domain to be the proposal of a step in the plan under construction. Similarly, solution strategies are equated with introducing new subgoals in the domain. For instance “Next we should bring the engine to Elmira" is a step, whereas "We need to decide which engine to use" is a strategy. As in tutorial dialogue, solution steps can be newly proposed or augmented. A step can be tabled, or hypothetically proposed for discussion. **Provide domain content** again refers to domain content which is not a step in the plan. When a step is addressed it can be evaluated, accepted, rejected or verified, and thereby adopted. The set of request types is similarly incomplete, but at least includes requests about domain objects, the state of affairs, and requests for a proposal of the next plan step.

In summary, we have prepared our taxonomy of dialogue moves for tutoring by further specifying the **Task** and **Task management** categories of the original DAMSL taxonomy. We have tried to keep as close to the DAMSL specification as possible with regard to the general dialogue level function, while adapting it to capture the phenomena of tutorial dialogue. We have presented the taxonomy at an abstract level, and shown how it can be instantiated not only to tutorial dialogue, but also to problem solving dialogue.

4 Validating the taxonomy

We first used the taxonomy to perform a small-scale annotation experiment on the validation set taken from the two theorem proving corpora introduced in Section 2. The data had previously been segmented into utterances. The goal of this experiment was to see whether our categorisation can be reliably applied to data and to validate the coverage of the taxonomy. The annotation was carried out by two annotators (the authors of this paper), following the definitions of the dialogue moves informally presented above. We did not consider the category information-relation because no definition is given by the original DAMSL taxonomy, however we will return to the question of information relation later in the discussion. Inter-annotator agreement is calculated using Cohen's kappa (Cohen 1960) on a per-dimension basis and the results of the experiment are as follows:

Dimension	κ value
Forward	0.87
Backward	0.47
Task	0.75
Task Management	0.91

These results can be considered very good for the forward dimension and the task management category, good for the task category, and low for the backward dimension. Among the categories with the lowest agreement were **Neutral** at 0.11 and **Solution step augmentation** at 0.37. In this preliminary evaluation our strategy was not to use a category "other" for utterances which did not appear to belong to any existing category, but rather to try to fit the annotation to the categories as they are.

Utterance	Forward	Backward	Info-level
S1	assert		task:contr:perform:new soln step
T1	assert	accept	task:addr:eval:correct
S2	assert		task:contr:strategy:state-future-step
T2		neutral	task:addr:hint
S3	assert	neutral	task:contr:perform:new soln step
T3	info-request	request-clar	task:req:explanation
S4	info-request	neutral	
T4	info-request	neutral	

Figure 2. Annotated example from the theorem proving domain

For the second part of our validation we now give examples of annotated data from each of the three datasets introduced above.

In Figure 2 we give the annotation of the excerpt from tutorial dialogue on theorem proving from Figure 1. It illustrates some of the types of problematic utterances which the corpora contain. For instance the utterances “Yes?” is a question and could appear to be **Information requests**, but in fact acts more like a prompt to continue, for which we had no category. Similarly the functions of the questions in sequence are difficult to abstract. We have tagged these as **Neutral**, since they discharge the obligations introduced by the questions before them, but the link between consecutive interrogative utterances is elusive.

We additionally annotated examples from the differentiation and collaborative planning corpora. Figure 3 gives the information level annotation for these examples. Our annotation of the dialogues on differentiation exhibits largely the same phenomena as the theorem proving data. There are more step augmentations, which may be due to the computational nature of the tasks. The short exercise length also leads to more task management utterances being performed. In the collaborative planning examples, the nature of the task leads to many more references to the state of affairs, for instance in order to communicate properties of objects when knowledge is not shared equally. The dialogues have more of a discussion character, shown by the higher number of proposals being tabled for discussion rather than performed directly.

Discussion We will now briefly discuss the results and findings of our annotation experiment and allude to some of the possible causes of the difficulties we encountered.

We believe that tutorial dialogue is inherently difficult to annotate reliably in a detailed way. One of the reasons for this is that students tend to be very concise, which makes it difficult to determine how they intended to relate the latest input to the previous discourse. This is reflected in our agreement score for the backward dimension, which at 0.47 is much lower than the other dimensions, as well as in

Utterance	Info-level
Tutorial Dialogue: Differentiation	
T: try this: $y = 1/(6x^2 - 3x + 1)$	taskmng:start
S: $dy/dx = -12x + 3/(6x^2 - 3x + 1)$	task:contr:perform:new soln step
T: Bracket problem again	task:addr:eval:incorrect
...	
S: $dy/dx = (-12x + 3)/(6x^2 - 3x + 1)$	task:contr:perform:augment step
T: Good	task:addr:eval:correct
and now what about the power of $(6x^2 - 3x + 1)$?	task:req:req augment step
...	
T: yes well done	task:addr:eval:correct
T: time to stop	taskmng:finish

Collaborative Planning	
S: There are actually two engines at Elmira	task:contr:non-step:soa
U: right	
S: Which one would you like to use	task:req:reqinfo:object
U: Number three	task:contr:perform:new soln step
S: OK	task:addr:accept
U: Might as well just take a boxcar from there as well	task:contr:perform:augment step
S: OK	task:addr:accept

Figure 3. Annotated examples from differentiation and collaborative planning

the agreement score of 0.37 for the **Solution step augmentation** category, which is heavily dependent on previous context. This result may even point to a general characteristic of tutorial dialogue which makes computational modelling challenging. In particular the **Neutral** category resulted in conflicting annotations because it is often unclear, as in the examples shown above, whether requests are being answered or merely addressed.

We have found that tutors typically perform utterances which contribute to many different goals — for instance they can simultaneously reject proposed solution steps while giving hints on how to continue in the task. The purpose of multidimensional dialogue move taxonomies is to handle this very multifunctionality, and while this is successful to a point, conflicts in the annotation experiment have highlighted some dual functions within the same category.

At least three categories have emerged that may need to be added to the current taxonomy to make it cover tutorial dialogue more completely. As discussed above, a prompt type in the forward dimension seems necessary. In addition, we would foresee a backward category which corrects a previous utterance, a category in the solving task which requests the next step in the solution, and a subcategory in task management to check if the current task is being restarted. Similar categories are proposed by Tsovaltzi and Karagjosova (2004), and may be taken up. We can additionally draw attention to the fact that there are many interrelations between the dimensions which are not captured by our presentation of the taxonomy, and which may for instance be accounted for by introducing constraints on label combinations.

5 Conclusions and future work

In this paper we have presented a taxonomy of dialogue moves which captures task-related actions. We have shown that the taxonomy can be instantiated to cover tutorial dialogue data (with which it was developed) as well as collaborative planning data. We then detailed an annotation experiment which applied the taxonomy to a validation data set of tutorial dialogues and achieved good inter-annotator agreement. This preliminary study showed that we are able to cover the data well. We went on to apply the taxonomy to further examples from tutoring and, in the appropriate instantiation, to examples from collaborative planning. We did however find a number of systematic phenomena in the data, such as the problem of relating task level actions to the previous discourse, which are of particular importance for classifying tutorial dialogue actions.

Closely related to our work is a recent study by Porayska-Pomsta et al. (2008), who categorise task related student actions and tutor feedback in a investigation of student affect. A simpler taxonomy is presented by Marineau et al. (2000), which differs from our approach in that it was developed with the goal of automatic classification in an intelligent tutoring system. In a pedagogically motivated analysis of a corpus of tutorial dialogues on computer literacy, Graesser et al. (1999) categorise tutors' actions in order to propose a model of tutorial dialogue structure in which a phase of "collaborative improvement of the answer" takes place. The work presented here will support further computational development of Graesser's model as it offers a more fine-grained categorisation of task-level actions performed within the improvement phase.

In our future work we plan a larger scale annotation of a further test set from our corpora. While the annotation experiment detailed here admittedly can not be seen as being conclusive because it has been carried out by the authors rather than independent annotators, we nevertheless believe further annotation will confirm the tendencies found so far. One of the goals of our work is to inform the development of models for tutorial dialogue, and so with a view towards operationalisation of the dialogue moves in our taxonomy, we will work on an axiomatic formalisation of the dialogue moves. This can form important input into developing a plan-based model for tutorial dialogue.

Acknowledgements We thank Kaśka Porayska-Pomsta, Helen Pain and Manolis Mavrikis for providing us with excerpts from the LeActiveMath corpus.

References

Allen, James and Mark Core (1997). Draft of DAMSL: Dialogue Act Markup in Several Layers. *DRI: Discourse Research Initiative* University of Pennsylvania.

Austin, John L. (1955). *How to do Things with Words*. Oxford University Press, 2005, Second edition, William James Lectures.

Benzmüller, Christoph, Armin Fiedler, Malte Gabsdil, Helmut Horacek, Ivana Kruijff-Korbayová, Manfred Pinkal, Jörg Siekmann, Dimitra Tsovaltzi, Bao Quoc Vo, and Magdalena Wolska (2003). A Wizard-of-Oz Experiment for Tutorial Dialogues in Mathematics. In Vincent Aleven, Ulrich Hoppe, Judy Kay, Riichiro Mizoguchi, Helen Pain, Felisa Verdejo, and Kalina Yacef (eds.), *AIED2003 Supplementary Proceedings*, volume VIII: Advanced Technologies for Mathematics Education, 471–481, Sydney, Australia: School of Information Technologies, University of Sydney.

Benzmüller, Christoph, Helmut Horacek, Ivana Kruijff-Korbayová, Henri Lesourd, Marvin Schiller, and Magdalena Wolska (2006). DiaWozII – A Tool for Wizard-of-Oz Experiments in Mathematics. In *Proceedings of the 29th Annual German Conference on Artificial Intelligence (KI-06), Lecture Notes in Computer Science*, 4314, 159–173, Bremen, Germany: Springer-Verlag.

Benzmüller, Christoph, Helmut Horacek, Henri Lesourd, Ivana Kruijff-Korbayová, Marvin Schiller, and Magdalena Wolska (2006). A corpus of Tutorial Dialogs on Theorem Proving; the Influence of the Presentation of the Study-Material. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-06)*, 1766–1769, Genoa, Italy: ELDA.

Buckley, Mark and Magdalena Wolska (2008a). A Grounding Approach to Modelling Tutorial Dialogue Structures. In Jonathan Ginzburg, Pat Healey, and Yo Sato (eds.), *Proceedings of LONDIAL 2008, the 12th Workshop on the Semantics and Pragmatics of Dialogue*, 15–22, London, UK.

Buckley, Mark and Magdalena Wolska (2008b). A Classification of Dialogue Actions in Tutorial Dialogue. In *Proceedings of COLING 2008, The 22nd International Conference on Computational Linguistics*, Manchester, UK.

Bunt, Harry (2000). Dialogue Pragmatics and Context Specification. In H. Bunt and W. Black (eds.), *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics*, volume 1, 81–150.

Callaway, Charles B. and Johanna D. Moore (2007). Determining Tutorial Remediation Strategies from a Corpus of Human-Human Tutoring Dialogues. In *Proceedings of the 11th European Workshop on Natural Language Generation*, Schloss Dagstuhl, Germany.

Cohen, Jacob (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20(1):37–46.

Graesser, Arthur C., Katja Wiemer-Hastings, Peter Wiemer-Hastings, and Roger Kreuz (1999). AutoTutor: A Simulation of a Human Tutor. *Cognitive Systems Research* 1:35–51.

Heeman, Peter A. and James F. Allen (1995). The TRAINS 93 Dialogues. Technical report, University of Rochester, NY, USA.

Marineau, Johanna, Peter Wiemer-Hastings, Derek Harter, Brent Olde, Patrick Chipman, Ashish Karnavat, Victoria Pomeroy, Sonya Rajan, and Art Graesser (2000). Classification of Speech Acts in Tutorial Dialogue. In *Proceedings of the Workshop on Modeling Human Teaching Tactics and Strategies, ITS 2000*, 65–71.

Porayska-Pomsta, Kaśka, Manolis Mavrikis, and Helen Pain (2008). Diagnosing and Acting on Student Affect: the Tutor's Perspective. *User Modeling and User-Adapted Interaction* 18(1-2):125–173.

Tsovaltzi, Dimitra and Elena Karagjosova (2004). A View on Dialogue Move Taxonomies for Tutorial Dialogues. In Michael Strube and Candy Sidner (eds.), *Proceedings of 5th SIGdial Workshop on Discourse and Dialogue*, 35–38, Cambridge, Massachusetts, USA: Association for Computational Linguistics.

Wolska, M., B. Q. Vo, D. Tsovaltzi, I. Kruijff-Korbayova, E. Karagjosova, H. Horacek, M. Gabsdil, A. Fiedler, and C. Benzmüller (2004). An Annotated Corpus of Tutorial Dialogs on Mathematical Theorem Proving. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-04)*, 1007–1010, Lisbon.

Visualization of dialect data

Erhard Hinrichs and Thomas Zastrow

Abstract. The field of dialectology relies on lexical knowledge in the form of pronunciation and lexical data. The present paper focuses on the recently developed approach of computational dialectometry, particularly on the scientific visualization techniques that have been developed within this approach. Existing visualization software packages are mature enough to automatically generate maps and other visualization formats in a matter of minutes. This provides the basis for a more in-depth analysis of the primary data. Electronic archives offer the possibility of including scientific visualizations of the analytic results in conjunction with the primary data and the raw output of the analytic algorithms used. This in turn enables other researchers to replicate the results or to use the primary data for further analysis.

1 Introduction[1]

Dialectology, the study of language variation, has enjoyed a long and venerable tradition in linguistics. Its practicioners have developed and adhered to a common methodology of data elicitation, data categorization, and ways of data visualization (see Chambers and Trudgill (1980) for an excellent introduction to the field). Data collection primarily relies on elicting pronunciation and lexical data from local informants from different areas of the territory under consideration. Data analysis aims at identifying isoglosses, i.e. geographic boundaries between characteristic linguistic features such as pronunciation of vowels, inflectional forms, or choice of lexical item to refer to particular objects of everyday life. Bundles of isoglosses form the basis of identifying dialect areas such as the *Benrath line* which distinguishes High German from other West Germanic languages. The results of such investigations are then typically charted in the form of geographic maps that display isoglosses or dialect areas. It should be obvious that producing high-quality maps is an extremely time-consuming and expensive process, leading to high production costs and, thus, limited distribution of the resulting publication. Moreover, book publication also imposes a clear limitation on how much material can reasonably be included, thus forcing researchers to make a selection amongst the available data.

[1] This work was carried out as part of a collaborative research project “Measuring linguistic unity and diversity in Europe” funded by the Volkwagen-Stiftung. We are extremely grateful to Hans Goebl for making his VDM software available to us and for extended discussion on scientific matters related to the research reported here. We would also like to thank three anonymous reviewers of the Konvens 2008 conference for their detailed and helpful comments on an earlier version of this paper.

Data storage typically took the form of index cards or analog data recordings, raising serious issues of data preservation and access. To mention just two examples: the enormous holdings of the Wenker Sprachaltas are currently being digitized in a costly and time-consuming research project. The Faculty of Slavic Philologies at the St. Kliment Ohridski University of Sofia has a collection of more than one million index cards, collected over the course of the last fifty years by fieldworkers studying Bulgarian dialects. Due to a lack of resources, it is to be feared that these data will eventually be lost.

Beginning in the 1970s, a new approach to dialectology was established by the pioneering work of Séguy (1971) and Goebl (1982). Goebl applied statistical methods to measure distances to abstract and visualize a basic pattern of similarities and dissimilarities among the large amounts of data found in the language atlases, particularly for Romance languages. Since the techniques used by Goebl and others involve quantitative measurements, this new approach to dialectology is aptly referred to as *dialectometry*. An even more recent development, starting in the 1990s is the field of computational dialectometry which applies methods from pattern recognition and unsupervised learning to measure dialect distances (see John Nerbonne and Kleiweg 1999; Kessler 1995).

Along with establishing a new methodology for dialectology, the field of dialectometry has developed entirely new methods for data representation and data visualization, which have the potential of revolutionizing the field of dialectology by paving the way for new interdisciplinary research between linguists, sociologists, and geographers. The results obtained by computational dialectometry normally take the form of similarity or distance matrices. Without proper visualization, such matrices are difficult, if not impossible, to interpret.

The focus of this paper is to explore in some depth the enormous potential offered by new means of visualizing scientific results in dialectology. The scientific results themselves are obtained by the computational techniques of dialectometry. The scientific results themselves are obtained by the computational techniques of dialectometry.

In this paper these techniques are presupposed and will be described only to the extent necessary for the kinds of visualization examples presented.

The context of the research reported here is a joint research project on Bulgarian dialects (hence-forth, referred to as the *Buldialects Project*[2]). The data collected in this project form the basis for the visualization techniques presented in the paper. However, the techniques as such are language independent and can be used for any language family or dialect area.

[2] http://www.sfs.uni-tuebingen.de/dialectometry

2 The Bulgarian data set

In cooperation with the Bulgarian Academy of Sciences and the University of Sofia, two data sets of Bulgarian dialects have been compiled:

- a set of phonetic data collected from 197 geographical distinct and representative sites across the Bulgarian territory. This collection contains a total number of 156 distinct word forms and their pronunciations.

- a set of lexical data collected from the same 197 locations with 112 lemmas, which exhibit char-acteristic geographic variation.

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

3 Dialectometrical workflow

A typical workflow in computational dialectometry contains three steps.

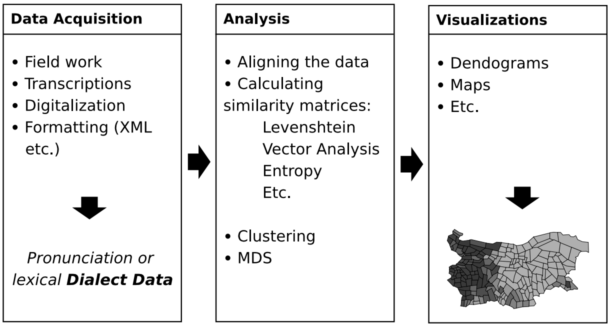


Figure 1. Workflow in Computational Dialectometry

This workflow has also been followed in the Buldialects Project. The field work was completed prior to the project, and transcriptions were available in the form of index cards. In a first step, this data was digitized and stored as IPA conformed X-Sampa encodings. Adhering to commonly agreed upon encoding standards ensures sustainability of the data and facilitates accessibility for other researchers in the field. To this end the project will make its primary data available by a web service that includes the primary data and the analysis and visualization tools developed in the project.

Data analysis proceeds in three steps: first, the data are organized in a two-dimensional matrix which provides a consistent ordering of the word forms across the different sites. This forms the basis for calculating the dialectometric distance between sites, for which different algorithms can be used, including Edit Distance Algorithms (John Nerbonne and Kleiweg 1999), Numerical Taxonomy (Goebl 1982), and Vector Analysis (Hinrichs and Zastrow 2007). These calculations result in a similarity or distance matrices, which in turn provides the input for partitioning the data into dialect clusters.

In a third step, the clustered data can be visualized in various ways. The remainder of the paper will concentrate on this third step.

4 Visualization

Traditional dialectology can only describe the dialect regions of a language or, at best, draw maps manually along isoglosses. In computational dialectometry, the use of quantitative methods allows the visualization of dialect regions in an automatic fashion and with much more fine-grained resolution in the form of maps or dendrograms.

The basis for all map-based visualizations is a *silent map* (in German: *stumme Karte*) of the entire region under consideration. Such a silent map shows only the territorial borders and possibly topographical information. On the basis of such a map, the dialect data can be visualized in two ways:

- As colored points, representing the individual sites
- As *Voroni maps*, dividing the whole area into smaller areas around the sites (Figure 2 is such a Voroni map)

There are two mature software packages available for visualizing dialect data: the L04 software[3], written by Peter Kleiweg at the University of Groningen, and the VDM software[4] , written by Edgar Haimerl at the University of Salzburg. The examples in this paper are generated with the VDM software.

Figure 2 visualizes the pronunciation data of the Buldialects project on the basis of a similarity matrix calculated by a vector chain analysis for the vowel “e”. This vowel was chosen because it is known to be highly indicative of dialectal differences of Bulgarian Gutschmidt (2002).

[3] http://www.let.rug.nl/~kleiweg/L04/

[4] http://www.sbg.ac.at/rom/people/proj/dm/vdm/features.html

The vector chain method allows analysis of single or of groups of aggregated X-Sampa codes in the dialect data. An X-Sampa element is traced through the dialect data by building a vector chain from its first to its last appearance. The vector chain forms a site-specific “fingerprint” showing the number of appearances of the X-Sampa code in focus, and its relative position changes through the data. Comparing these fingerprints for several sites gives evidence for the dialectal differences between them. For more details, see Hinrichs and Zastrow (2007).

Figure 2 was automatically generated as a synopsis map by the VDM software[5]. A synopsis map requires the choice of several parameters, including a predefined number of distinct classes. In the case of Figure 2, ten distinct classes were chosen.

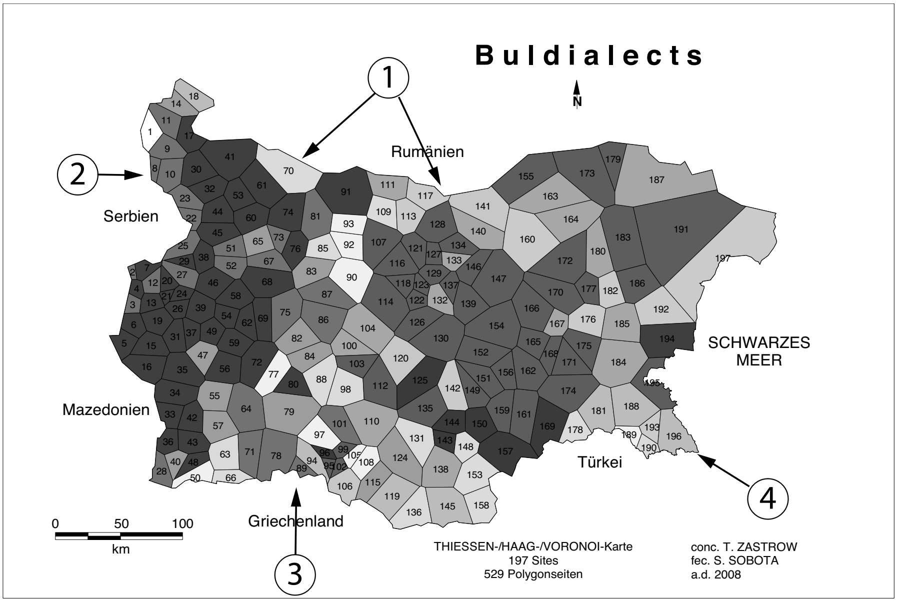


Figure 2. Automatically generated synopsis map

The map shows the following characteristics of Bulgarian dialects. For ease of reference they are highlighted by numbered arrows:

1 The so-called *jat line*, named after the presence or absence of the semi-vowel *j*, dividing the eastern and western part of Bulgaria by a line running from the north in a southwestern direction.

[5] The so-called synopsis map was used with site no. one as reference point. For a detailed view, see `http://ald.sbg.ac.at/dm/Engl/default.htm` and Goebl (1982)

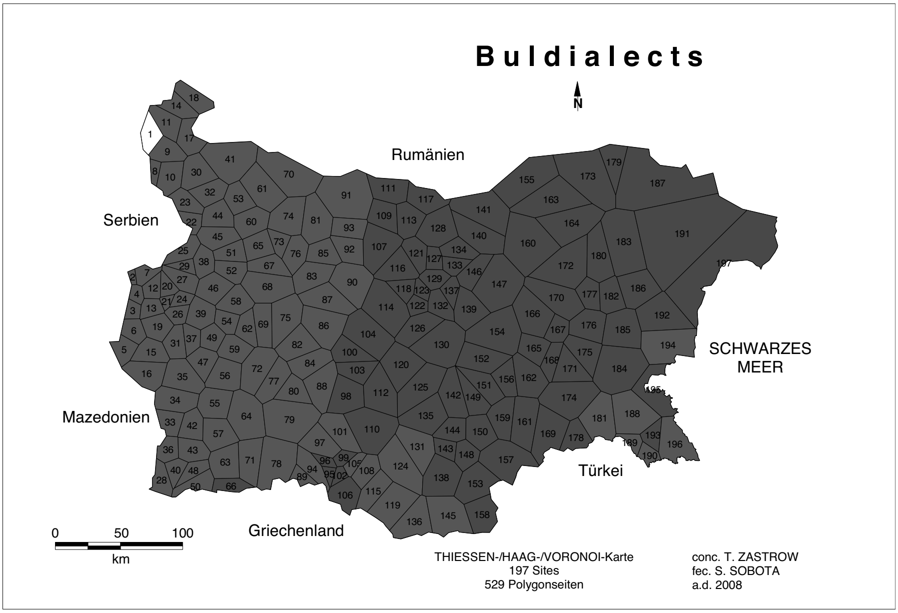
2 On the border to Serbia and north of Sofia, there is a belt of transition dialects between Bulgarian and Serbian.

3 In the south, the mountains of the Rhodopes form a separate and heterogeneous dialect region. In the middle of that segment, there is a break towards Plovdiv (dark blue).

4 The sites at the border to Turkey form a region of their own.

Interestingly, all of these findings are in full agreement with the results of traditional Bulgarian dialectology.

It is important to note that the synopsis map in Figure 2 represents only one possible parameter setting[6]. Typically an analyst will want to compare a large number of such maps and select the map that shows the right kind of resolution for a particular dialectal feature. For the synopsis map in Figure 3, only two classes were chosen as a predefined parameter so that the east-west dividing line as the main characteristic feature of Bulgarian dialects is highlighted.



THIESSEN-/HAAG-/VORONOI-Karte
197 Sites
529 Polygonseiten

conc. T. ZASTROW
fec. S. SOBOTA
a.d. 2008

Figure 3. Two classes, showing the east-west distinction of Bulgaria

[6] In Figure 2, site 1 was used as *Reference Point* for all sites

Since all VDM maps can be generated automatically in a matter of minutes, there are no practical limitations as to the number of maps that can be considered and included in an electronic archive. However, for the traditional publication format of a book, the number of maps to be included is severely limited. From a practical as well as from a scientific perspective, an electronic archive is therefore clearly the way of the future in a traditional humanities field such as dialectology. Computational dialectometry therefore introduces not only novel analysis techniques, but also an entirely new form of dissemination of scientific results.

4.1 Other kinds of visualization

Another common kind of visualization is hierarchical clustering. Different clustering algorithms[7] produce slightly different visualizations. In addition to geographical maps, dendrograms display the differentiation of clusters (Figures 4 and 5).

Other possibilities of visualization are isogloss (Figure 6) and ray maps (Figure 7). Isogloss maps display dialect structures as more or less strong borders between dialect regions. The thickness of the line between two regions indicates the degree of dissimilarity between the separated regions. Ray maps are the opposite of isogloss maps: they show the communication and not the border between dialects. The thickness of the line between two regions indicates the degree of similarity between the separated regions.

5 Conclusion and future work

Computational dialectometry introduces not only novel analysis techniques into the traditional field of dialectology, but also an entirely new form of dissemination of scientific results. Existing visualization software packages are mature enough to automatically generate maps and other visualization formats in a matter of minutes. This provides the basis for a more in-depth analysis of the primary data. Electronic archives offer the possibility of including scientific visualizations of the analytic results in conjunction with the primary data and the raw output of the analytic algorithms used. This in turn enables other researchers to replicate the results or to use the primary data for further analysis.

We believe that the innovative potential of computational dialectometry is far from exhausted. One of the recurring themes in the study of dialectology concerns the influence of topography and other geographic as well as demographic factors

[7] For a detailed explanation of clustering algorithms, see im Walde (2003)

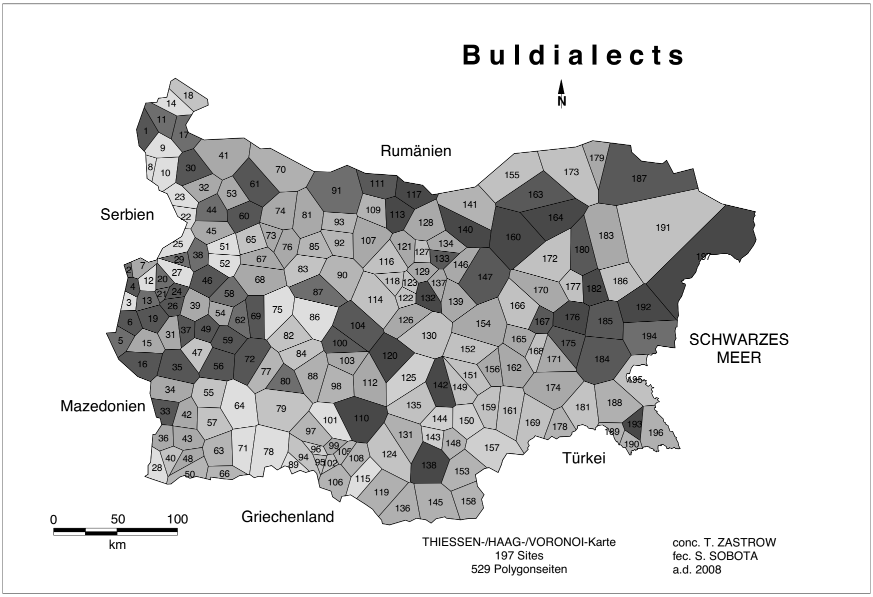


Figure 4. Clustering with the WARD method, 13 clusters

on language variation. The increased availability of geographic and demographic data in conjunction with the new visualization techniques for dialectology, which we have described in some detail in this paper, will make it possible to reconsider such questions in a much more systematic way. More specifically, the dialect data can be combined with other - geographical - data in the form of layers. Such layers can contain information about roads, mountains, rivers, and so on. Toggling these layers on and off would allow a differentiated look on the mutual influence between dialects and other factors (Figure 8 shows three possible layers).

In combination with a *Geographic Information System* and a *Mapserver* it is possible to publish huge amounts of maps and the corresponding dialect data as a web service. Depending on the available software and its configuration, it also should be possible to visualize dialect data *on the fly*.

References

Chambers, J. K. and Peter Trudgill (1980). *Dialectology*. Cambridge University Press.

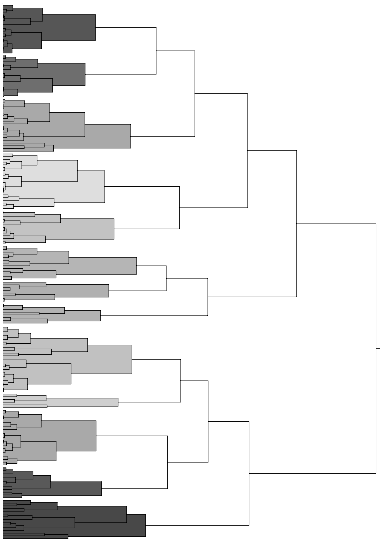


Figure 5. A dendrogram, corresponding to Figure 4

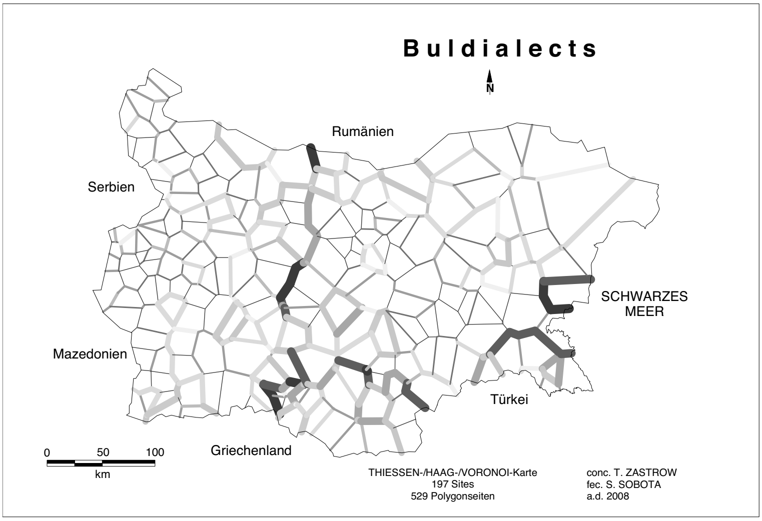


Figure 6. An Isogloss Map

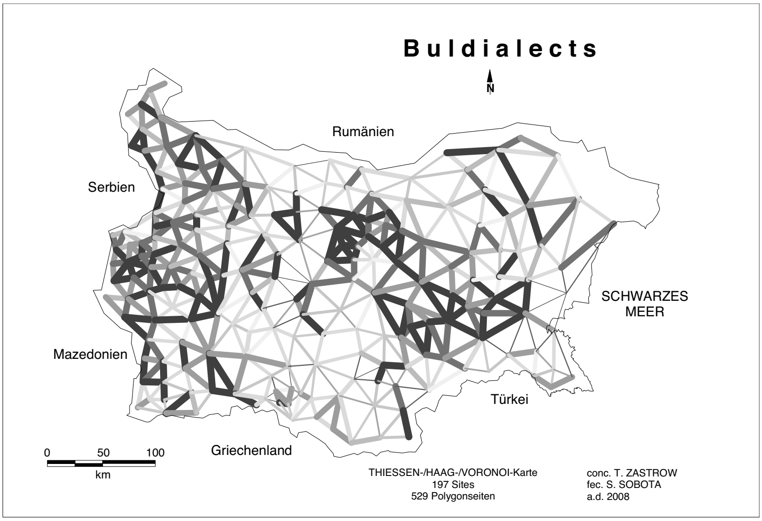


Figure 7. A Ray Map

Goebl, Hans (1982). *Prinzipien und Methoden des Einsatzes der Numerischen Taxonomie im Bereich der Dialektgeographie*. Wien.

Gutschmidt, Karl (2002). Bulgarisch. *Enzyklopädie des Europäischen Ostens 10* 219–234.

Hinrichs and Thomas Zastrow (2007). A Vector-Based Approach to Dialectometry. In *17th Meeting of Computational Linguistics in the Netherlands*.

John Nerbonne, Wilbert Heeringa and Peter Kleiweg (1999). *Edit Distance and Dialect Proximity*.

Kessler, Brett (1995). Computational Dialectology in Irish Gaelic. In *EACL-95*.

Séguy, Jean (1971). La Relation entre la Distance Spatiale et la Distance Lexicale. In *Revue de Linguistique Romane*, 35, 335–357.

im Walde, Sabine Schulte (2003). *Experiments on the Automatic Induction of German Semantic Verb Classes*. Ph.D. thesis.

Zhobov, Vladimir (2006). Description of the Sources for the Pronunciation Data, unpublished Manuscript. Department of Slavic Philologies, University of Sofia.

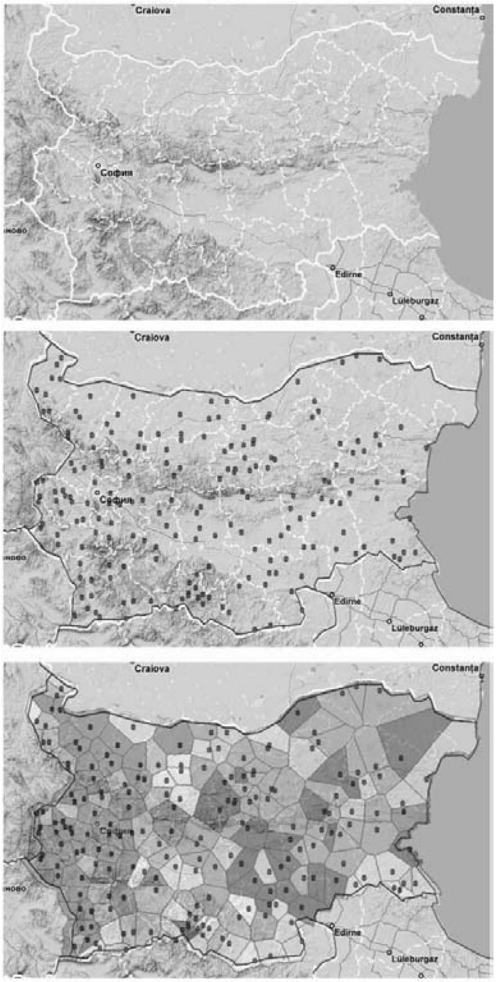


Figure 8. topographic map (top) , geographic sites superimposed as points (middle), additional superimposed Voroni map (bottom)

Providing corpus data for a dictionary for German juridical phraseology

Ulrich Heid, Fabienne Fritzinger, Susanne Hauptmann, Julia Weidenkaff and Marion Weller

Abstract. We report on a set of procedures to extract juridical terminology and in particular collocations and chains of collocations from German texts from the field of the protection of intellectual property. The extraction work is based on standard corpus technology, and it produces different types of output which can be made available to lexicographers: lists of term candidates, of collocation candidates, as well as example sentences which show the use of these items in context.

German juridical language is a particularly interesting domain to carry out analyses of multiword expressions, as it is very rich in collocations and in chains of collocations. Many of these multiword expressions correspond to juridical concepts and thus merit being extracted and described in an electronic dictionary.

We describe our sources, our tools, and examples of the output of the extraction procedures; and we discuss options for the presentation of these results towards lexicographers.

1 Introduction: objectives and tasks

Juridical discourse contains a large number of multiword expressions many of which can be seen as belonging to specialized phraseology in the sense of Picht (1987): these multiword expressions are specific to the given domain, and they are a conventional way to express specific concepts of the domain. Moreover, they are idiosyncratic in terms of lexical selection and thus need to be learnt by those who want to be able to use the juridical sublanguage of German. Examples of multiwords with these properties are given in section 4.

The main objective of the work described in the following is to extract both single word and multiword term candidates and collocation candidates from a large amount of textual data compiled from a collection of law texts and, in particular, from a juridical journal. The extracted material is to serve jurists and lexicographers in the preparation of a bilingual specialized dictionary , German/English, English/German, which will contain, in addition to equivalence data, a considerable amount of phraseology.

In this paper, we will discuss the sources used for the data extraction (section 2), the techniques applied, which are mainly relatively simple linguistic and lexicostatistical tools (section 3), as well as questions of the presentation of the resulting data towards lexicographers desiring to compile dictionary entries for domain-specific phraseology (section 4).

The functionality of the tools used for this project is roughly similar to that of the *sketch engine* (Kilgarriff et al. 2004), augmented by simple procedures for term candidate identification.

2 Sources of juridical texts and computational linguistic preprocessing

2.1 Text basis

The work described in this paper is the result of a cooperation between C. H. Beck publishers, München, and computational linguists form Stuttgart University. In the framework of this cooperation, the publishing house has made available to the university large electronic text collections . On the one hand, collection of laws and regulations has been used which concerns intellectual property rights, patent law as well as laws on trademarks and on (unfair) competition. These legislative texts contain a total of 440,000 tokens.

A second, much larger text collection has also been made available by the publishing house: the collection of almost 60 years of specialized journal articles from the same juridical subdomains as the laws, published in several German specialized journals. These include *GRUR (Gewerblicher Rechtsschutz und Urheberrecht), Neue juristische Wochenschrift – Entscheidungsdienst Wettbewerbsrecht* and *GRUR-RR (GRUR Rechtsprechungsreport).* This text collection contains a total of 78 million tokens, which is quite sizeable in comparison with typical sublanguage corpora. In experiments with English corpus data, a collection of texts from the IIC journal was used with a total of 11.2 million words.

2.2 Specificities of the source material

The quantitative dimension of the text material extracted from the specialized journals is of particular relevance for the extraction of specialized phraseology: in a larger corpus, obviously, more word combinations will show up with a sufficient number of occurrences for statistical tools (e.g. association measures) to provide significant results. As Evert (2004:133) suggests, the extraction of collocation candidates should not take into account word combinations which occur less than 5 times in a corpus.

When discussing the optimal corpus size for the identification of the core terminology of a scientific domain, Bergenholtz/Tarp (1995:94s.) suggest that one million words should be sufficient. This, obviously, presupposes a corpus carefully selected and composed of texts which are relevant and central for the domain under analysis, and which cover the most important aspects of this domain.

One could argue that also for collocation candidate extraction a “balanced” corpus would be much preferable. In the case of our collection of articles, this would however imply further subclassifying the articles from the specialized journal in order to arrive at an even coverage of the subdomains of the field of property rights. Unfortunately, such a subclassification is not available from the data made accessible by the publishers. And due to the complexity of the domain, it would be hard to achieve. This is why the opportunistic collection of all articles of the specialized journal was used in this case. And we assume that the quantitative dimension of the corpus levels out some of the deficiencies which may be caused by its composition.

Moreover, it is not easy to define the core of the field of property rights and trademark legislation, as the articles contained in the journal not only deal with juridical matters, but also with the contents-wise background of the respective laws and litigations. Examples of such cases include the use of the yellow colour used by the German postal services, as a trademark; or possibilities to turn nicknames into trademarks; or again unfair competition between pharmacies. In such cases, specialized terminology (and even ad hoc denominations) from the respective domains (e.g. pharmacy and trade) are used in the articles, when the individual cases are discussed. To counterbalance these effects, the full set of articles from the journal was used, on the assumption that word and word pair statistics would even out local bursts in individual texts (in fact TF/IDF or a burst analysis would give even more precise figures here).

There are also frequency effects in the data under analysis which are not related with collocations; in fact, general juridical adjectives, such as for example *streitgegenständlich*, can show up with a very wide range of nouns, as almost any concrete object or service can be the object of litigation. Examples include *streitgegensti"andliche Bettwäsche, streitgegenständliches Videospiel, streitgegenständlicher Film*, etc. None of these, despite their high frequency in single texts, merits being included in the dictionary. Often, filtering by means of association measures, such as the log-likelihood ratio test (cf. Dunning 1993), allows filtering of these combinations.

On the other hand, not all word combinations which are phraseologically relevant for the domain consist of highly specialized terms; the multiword expression *rein beschreibende Angabe* (“merely descriptive indication”) is a piece of specialized phraseology of trademark law as it denotes a particular way of textually describing the properties of a product (the contrary of comparative advertisement, which is prohibited by German law). This multiword expression is thus lexicographically relevant, even if it does not contain a single word which would not also appear in general language.

2.3 Metadata annotation

One of the objectives of our data extraction exercise is to provide lexicographers with corpus sentences which can serve as typical usage examples (see below, section 4.3). We want to illustrate single word terms in this way, but also multiword terms and collocations (this is in line with Heid/Gouws 2006 who propose to treat collocations as lexicographic treatment units, just like single word items). For lexicographers to be able to correctly situate and interpret the examples offered by the system, several types of metadata are needed:

- Ref_Zs: Title of the journal from where the sentence is extracted
- Ref_Heftjahr: Year of publication
- Ref_S: Page number
- Typ: Text type (e.g. law, article, jurisdiction, etc.)
- Gericht: Name of a court whose decision is being cited
- Aktenzeichen: Reference number of a case, decision, etc.

The metadata are part of the material provided by the publisher, as the texts made available are intended for browsing in a hyperlinked text collection. We only had to remove a few types of metadata which were not relevant for the linguistic analysis and to adapt the format to the corpus tools used. The short list given above contains the metadata which are usable as a query criterion and automatically given as part of the documentation of cited examples. As the corpus contains journal articles from a span of 60 years, and as regulations, interpretations and terminology may have changed during this time span, it is necessary to indicate the year of publication of each decision or article.

3 Data extraction: standard corpus linguistic techniques

Our data extraction word uses standard corpus linguistic tools and techniques: preprocessing, pattern-based extraction with the CorpusWorkBench tools, CWB (Evert 2005), a simple comparison of relative frequencies according to Ahmad et al. (1992), and a calculation of word pair significance with the UCS toolkit (Evert 2004). We summarize these steps briefly, in the following.

3.1 Preprocessing

The texts are tokenized and tagged for part of speech (according to the STTS tagset[1]) with TreeTagger (Schmid, 1994): To adapt the tagger to the domain, word forms that are not contained in the lemmatization lexicon are extracted beforehand, and analyzed with the morphological analyzer SMOR (cf. Schmid et al. 2004); this provides a frequency list of candidate pairs of word forms and lemmas to be added to the tagging lexicon. The candidates are validated or modified interactively and included in the tagger lexicon. This procedure considerably improves tagging quality. In a subsequent step, the texts are both chunked (with the YAC Chunker, Kermes 2003) and parsed with Schiehlen's (2003) dependency parser. We treat the collection of laws and the journals as separate corpora.

3.2 Extraction of single word term candidates

To identify single word term candidates, we rely on the well-known approach (cf. Ahmad et al. 1992) to compare relative frequencies of single items (lemmas, in our implementation) from the specialized texts with the relative frequencies of the same lemmas in non-specialized texts[2]. The comparison throws up two types of relevant results: (i) items only used in the specialized texts, and (ii) items which are found in both texts, but proportionally (much) more often in the specialized texts. The second group also covers everyday words which have a specific juridical meaning and use[3]. Manual evaluation has shown that items from group (ii) which are at least seven times as frequent in the specialized text as in non-specialized texts lead to acceptable term recall and to ca. 50% precision. We rely on the jurists and lexicographers to decide about inclusion of these items the dictionary; from the subset fulfilling criterion (i), we propose all candidates (down to a frequency threshold dependent on the overall target size of the dictionary).

To identify the core terminology of the field, we check for all term candidates from the juridical journal subcorpus, whether they are also part of the law subcorpus (see above, section 2.1), and we mark the result (+/-) on the candidates as an extra indication for the lexicographers.

[1] URL (as of July 2008): http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-table.html

[2] As a non-specialized test, we use the 40 million words extract of *Frankfurter Rundschau* 1992/93 distributed by the *European Corpus Initiative*.

[3] In a similar exercise on Dutch family allowance legislation, in fact *kind* ("child") was thrown up as one of the top-ranked term candidates, by this procedure.

3.3 Extraction of collocation candidates

3.3.1 Adjacent multiword expressions.

We use pattern-based extraction procedures (which rely on the CQP query language of the CorpusWorkBench, Evert 2005) to identify adjacent multiword candidates, such as combinations of nouns and their modifying adjectives, of nouns and their genitive or prepositional complements, adjectives and their adverbs etc. These search patterns are automated and used to collect data on lemma combinations and on the form (e.g. singular/plural) of the pairs (cf. Heid 1998).

3.3.2 Noun+verb-collocations.

German collocations with verbal components are harder to extract than, e.g. English ones, because of the different verb placement models of German (which lead to the need to use more different search patterns) and because of the high degree of case syncretism in German noun phrases (which reduces precision). To account for these problems, we use two types of procedures: (i) a chunking-based precision-oriented extraction (along the lines of Heid/Weller 2008) and (ii) a parsing-based extraction on the output of Schiehlen's (2003) dependency parser . We describe each procedure in turn.

The chunking-based approach maximizes precision by explicitly modelling the three German verb placement models (verb-initial sentences, verb-second and verb final), as well as active vs. passive voice. To counterbalance the case ambiguity problem, we use Eckle-Kohler's (1999) subcategorization lexicon to distinguish verb + subject and verb + object collocations. In addition, the extraction patterns allow us to identify the morphosyntactic properties of each collocation candidate, e.g. determiner use (definite, indefinite, null, etc.), number, etc. Compound heads are equally identified (*Auschlußfrist, Abgabefrist, … → Frist*), so as to be able to sort compounds on their heads. The distribution over active vs. passive voice is covered by the use of different extraction patterns, and passive auxiliaries (*werden* vs. *sein*) are extracted with the same devices as determiners.

We also extract adjectival and adverbial modifiers, as these are part of collocational chains. However, prepositional phrases found in the Mittelfeld of German sentences tend to be hard to subclassify into relevant vs. irrelevant, as complement PPs and adjuncts may appear anywhere in Mittelfeld. Wherever possible, we collect however also the PP modifiers found by the system. An example are the PP *im Patentgesetz* and the adverb *unmittelbar* in *die Wiederholbarkeit wird im Patentgesetz nicht unmittelbar gefordert.*

The observed morphosyntactic properties of each sentence analyzed are stored as features in a database (cf. Heid/Weller 2008) and added up for each lexical pair (verb and noun lemmas). Examples of the results obtained for morphosyntactically flexible collocations (cf. *Gebühr* + *entrichten*, "pay + fee") and for fixed ones (*den Gegenstand (von X) bilden*, "be the object (of X)") are given in the database dump in (3), below, with columns for frequency, noun and verb lemma, determination type, number and voice:

(3)

f	n_lemma	v_lemma	det_type	num	voice
37	Gebuehr	entrichten	def	Sg	aktiv
37	Gebuehr	entrichten	indef	Sg	aktiv
29	Gebuehr	entrichten	def	Pl	aktiv
3	Gebuehr	entrichten	def	Sg	passiv
3	Gebuehr	entrichten	null	Sg	aktiv
3	Gebuehr	entrichten	quant	Sg	aktiv
2	Gebuehr	entrichten	dem	Pl	aktiv
1	Gebuehr	entrichten	def	Pl	passiv
1	Gebuehr	entrichten	indef	Sg	passiv
1	Gebuehr	entrichten	quant	Pl	aktiv
200	Gegenstand	bilden	def	Sg	aktiv
25	Gegenstand	bilden	null	Sg	aktiv
4	Gegenstand	bilden	poss	Sg	aktiv
1	Gegenstand	bilden	quant	Sg	aktiv

In (4), we reproduce the full description automatically captured in the database for an individual sentence. Given its feature decoration, it can be selected to illustrate exactly a given combination of features: the analysis not only provides a summary of feature distributions, it also allows us to identify example sentences for any of the properties under analysis.

(4) Sample entry for *Vertrag rückgängig machen*

```
n_lemma       | Vertrag
v_lemma       | rueckgaengig machen
morph_head    | Vertrag
modal         | koennen
modif         | PP:unter:Voraussetzung, PP:von:Verkaeufer
det_type      | def
num           | Sg
cas           | Nom
chunk         | Der Vertrag konnte unter bestimmten
              | Voraussetzungen vom Verkaeufer rueckgaengig
              | gemacht werden
sent_type     | v-2
voice         | passiv
```

```
    pass_aux       | werden
```

Parsing-based Extraction.. To supplement our data on verb+noun collocations, we also use a parsing-based approach: we extract verb+object pairs from the output of Schiehlen's (2003) dependency parser FSPAR. The parser produces dependency structures with a grammatical function annotation and an underspecified representation of label ambiguities and of attachment ambiguities. Extraction in done by Perl scripts applied to this output representation.

In a preliminary experiment, we compared the collocation candidate lists from both approaches. Contrary to a widespread assumption, it is not as much precision, but rather recall which is enhanced through the use of parsed data. An analysis of the top-250 collocation candidates of each approach shows an overlap of over 90 %, with divergences explainable by technical details of the tools used. In other words: the chunking-based extraction procedures achieve a good precision (which can not be topped through the use of parsed data). However, the recall of the chunking-based approach is at most around 40 % of that of the parsing-based technique; the chunking-based queries are geared to precision, at the expense of recall; for example, due to the Mittelfeld word order problem, we do not extract collocation candidates from chunked verb-second active sentences, while we do from parsed ones. Moreover, FSPAR contains a subcategorization dictionary to identify verb+object collocations.

>From the chunked GRUR journal data, we extract 958.678 verb+noun collocation candidate tokens; from the parsed data, we get 1.496.401 such candidate pair tokens. The same holds for collocation candidate pair types: 254.930 by way of chunking vs. 535.098 by parsing.

As the parser marks case ambiguities not resolvable in the sentence context (in our data around one third of all cases), we will need to check, in future work, whether and how precision and recall of the collocation candidate extraction depends on the use of unambiguously identifiable object NPs. There are a few collocations where the same lexical items show up in different grammatical patterns, and where these patterns belong to different collocations: *Frage stellen* ("ask + question") vs. *Frage stellt sich* ("question + arises") vs. *in Frage stellen* ("put into question"). We are also working towards the full use of morphosyntactic features in collocation extraction from the parsed data to account for such cases.

4 Examples of juridical phraseology and its representation for lexicographers

4.1 Examples of juridical phraseology

The analysis of specialized text supplies collocations belonging to general juridical language, but also collocations that originate from the more restricted domain of the protection of industrial intellectual property. Several different collocation types can be distinguished, e.g. noun+verb collocations (cf. the examples in (5)) or adjective+noun collocations (cf. (6)). The collocations in (6) also show variation of the nominal term.

(5) Noun+Verb Collocations (absolute frequency, N, V):

```
721 | Berufung                   | einlegen
599 | Beschwerde                 | einlegen
297 | Einspruch                  | einlegen
283 | Widerspruch                | einlegen
160 | Revision                   | einlegen
118 | Rechtsbeschwerde           | einlegen
115 | Rechtsmittel               | einlegen
...
 28 | Anschlussberufung          | einlegen
 11 | Anschlussrevision          | einlegen
 10 | Verfassungsbeschwerde      | einlegen
 10 | Sprungrevision             | einlegen
```

(6) Adjective+Noun Collocations incorporating the adjective *strafbewehrt* (nouns displayed in decreasing frequency order)

Unterlassungserklärung, *Unterlassungsverpflichtungserklärung*, *Unterlassungsverpflichtung*, *Unterwerfungserklärung*, *Verpflichtungserklärung*, *Unterlassungsvertrag*,...

4.2 Collocation chains

The use of verb+adverb collocations and of collocations that consist of verbs occurring together with predicative adjectives seems to be characteristic for juridical terminology: *etw. abschließend/endgültig/höchstrichterlich klären* (cf. (7) and (8)). Many of these combinations have term status[4] (i.e. the phrases denote legal realities), even though precisely these combinations are often missing in (specialized) dictionaries.

[4] The examples in (7) even show the technicality of the collocations: *unrechtmäßig handeln* is notably less frequent than e.g. the more specialized collocation *schuldhaft handeln*. In contrast, the adverb lemma *unrechtmäßig* occurs 235 times (in combination with different verbs) in newspaper text of about 150 milion words (*Stuttgarter Zeitung*, *Frankfurter Allgemeine Zeitung*, *Frankfurter Rundschau*) compared to 67 occurrences of the more technical adverb *schuldhaft*. Collocations with *handeln* do not occur at all in the newspaper texts.

(7) Verb+Adverb/Adjective Collocations (ADV/ADJ – verb – absolute frequency)

```
schuldhaft              | handeln | 63
wettbewerbswidrig       | handeln | 46
fahrlaessig             | handeln | 33
unlauter                | handeln | 22
rechtswidrig            | handeln | 18
vorsaetzlich            | handeln | 16
rechtsmissbraeuchlich   | handeln |  5
boesglaeubig            | handeln |  5
ermessensfehlerhaft     | handeln |  2
unrechtmaessig          | handeln |  2
```

The combination of noun+verb collocations with verb+adverb collocations leads to (even statistically) significant collocation triples, i.e. collocation chains: each of the nouns in (8) may combine with the respective verbs and take one of the adverbs[5].

(8)

Klage *Beschwerde*	*als unbegründet* *als unzulässig*	*abweisen*
Frage *Punkt* *Rechtslage*	*abschließend* *höchstrichterlich*	*klären*
Berufung *Beschwerde* *Rechsmittel* *Widerspruch*	*fristgerecht* *fernschriftlich* *(rechts)wirksam* *ordnungsgemäß*	*einlegen*

In the examples of (8), the verb is the collocate of the noun and at the same time it constitutes the basis of the collocation with the adverb (cf. also Hausmann 2004). The chains are thus the result of a recursive process.

4.3 Presentation for lexicography

Material. Plenty of collocation candidates of different structures can be extracted from the available data set. Nouns can combine with three different collocation partners (namely adjectives, nouns or verbs), to build up to five different collocation structures: adjective+noun, noun+verb, where the noun is either the subject or the object of the verb, and noun+noun, either as head of a genitive NP or as its genitive attribute. An outline of the data collected for the noun *Tatbestandsmerkmal* is given in (9)[6].

[5] Cf. Zinsmeister/Heid 2003 for the extraction of such phrases from standard German.

(9) Data Material for *Tatbestandsmerkmal*:

```
Tatbestandsmerkmal (NN)
ADJ:     ungeschrieben, subjektiv, objektiv, gesetzlich, erfuellt
NNgen1: T. der Vereinbarung, T. der Unlauterkeit, T. der Sittenwidrigkeit
NNgen2: Auslegung des T., Vorliegen des T., Erfuellung des T., Pruefung des T.
VVobj:  T. erfuellen, T. verwirklichen, T. auslegen, T liegt vor;
```

Besides the raw listing of collocation candidates, example sentences for each multiword item are automatically extracted from the text collection. To enable an exact citation of sentence extracts (if required), only those sentences are retained, where all metadata[7] are specified, i.e. *type* of text, *court* reference, *file number* and finally, a specification of the *journal* the sentence is taken from. Out of all sentences that meet these conditions, three to five sentences of medium length are selected in the end. One such example sentence is given in (10), along with its metadata:

(10)

```
Typ: Rspr        Gericht: LG Duesseldorf  Az: 38 0 39/04
ZS: GRUR-RR      Jahr: 2005               Seite: 96

Die einstweilige Verfuegung vom . . war daher mangels Verfuegungsanspruch
aufzuheben und die Antraege auf Erlass einer <einstweiligen Verfuegung>
zurueckzuweisen.
```

Methodology – Tools for Presentation. The production process of the dictionary itself is on the one hand based on the data retrieved with the procedures discussed, but on the other hand still depends on the domain expertise of the author and on the dictionary editing tools he uses.

The data provided by the automatic extraction procedures must be reviewed by the lexicographers. To avoid a material overload, a stepwise strategy is tested: the lexicographer first decides about the single word terms for which dictionary entries shall be created, and only then, in a second step, takes phraseology into consideration and makes a decision about which collocations are added to the single word terms to enrich the dictionary entries.

However, sometimes only the contexts in which a single word term occurs make it worth to be included in the dictionary. This might be the case when e.g. a term occurs in contexts that are unexpectedly frequent or particularly deviant with respect to the standard language. In this case, the stepwise approach is not sufficient and a different strategy should be favoured, in which all collocation candidates are simultaneously at the author's disposal. The data must then be clearly arranged for the author, for

[6] An example sentence for the collocation *Tatbestandsmerkmal der Vereinbarung* is: "Die Parteianträge stützen sich insbesondere darauf, daß ..., wenn das Urherberrecht selbst als gesetzliche Regelung nicht von den <Tatbestandsmerkmalen der Vereinbarung> oder Abstimmung in Art. 86 EWG-Vertrag erfaßt werde, so doch die Ausübung des Urheberrechts Objekt, Mittel oder bloße Folge einer Vereinbarung sein könne...".

[7] See section 2.3 for details.

example in the way this happens in the interface of the *Sketch Engine* (Kilgariff et al. 2004).

Such synoptic presentations often discard term variation (cf. example (6) above), and abstract away from morphosyntactic peculiarities, but there are GUIs in which such details can be inspected (through click or mouse-over), if required. To enable efficient lexicographic work, an application should not only display the collocations, but also provide the possibility selecting the ones that are to be included in the dictionary, as it happens e.g. in LexiView (cf. Heid et al. 2004). In addition, the lexicographer should be able to influence (e.g. through numbering) the order in which the selected collocations appear in a prototypical dictionary entry (which is generated in accordance with the stylesheet of the respective dictionary to be built). The arrangement of collocations could also happen automatically: they may be grouped according to their internal grammatical structure and then, on a deeper level, sorted either according to the alphabet, their absolute frequency or associative strength. In our studies, we decided to use a default sorting according to associative strength in terms of the log likelihood ratio.

5 Conclusion and future work

We presented a procedure for the extraction of specialized phraseology from text corpora and illustrated it with the task of identifying collocations and collocational chains in the terminology of intellectual property protection. We use generic components for preprocessing (tagging, lemmatization, chunking/parsing), for pattern-based search (CWB) and for matching terminological texts against standard texts (following Ahmad et al. 1992). We interconnect these components via scripts to a pipeline and produce output for the lexicographer. Different degrees of detail can be selected: candidate lemmas for the terminological dictionary, collocation candidates and example sentences including references.

The pipeline can be adapted to comparable instruments for other languages. Besides exchanging the preprocessing components, we only had to adjust some of the search patterns to get an English version to work.

The tools are not domain-dependent; however, it seems that in particular juridical phraseology is prone to build collocation chains (see section 4.1). The presence of huge amounts of text (in our case 75 million words of domain text) is highly advantageous for the extraction and analysis of such chains.

Future work aims at the more detailed investigation of such chains (Which part-of-speech patterns do they consist of? How restricted is the lexical inventory?) and at improved procedures to extract relevant instances that occur with low frequency. The data set can also be used to elaborate correct and precise procedures for the comparison of values for relative frequency or significance of collocations originating from specialized language vs. general language.

In future, we will rely exclusively on FSPAR (Schiehlen 2003) for collocation extraction and extract the relevant morphosyntactic preferences of the collocations from parsing output. This will also allow an extensive evaluation (in terms of precision and recall) of the advantages and disadvantages to use parsing-based methods as opposed to chunking-based procedures when extracting collocations of different types (e.g. adjective+noun, verb+subject, verb+object and verb+prepositional object).

References

Ahmad, Khurshid, Andrea Davies, Heather Fulford, and Margaret Rogers (1992). What is a Term? The Semi-Automatic Extraction of Terms from Text. In Mary Snell-Hornby et al. (ed.), *Translation Studies – an Interdiscipline*, Amsterdam/Philadelphia: John Benjamins Publishing Company.

Bergenholtz, Henning and Sven Tarp (eds.) (1995). *Manual of Specialised Lexicography – The Preparation of Specialised Dictionaries*. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Dunning, Ted (1993). Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics* 19(1):61–74.

Eckle-Kohler, Judith (1999). *Linguistic Knowledge for Automatic Lexicon Acquisition from German Text Corpora*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, Logos, Berlin.

Evert, Stefan (2004, published 2005). *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, available from `http://www.collocations.de/phd.html`, software: `http://www.collocations.de/software.html`.

Evert, Stefan (2005). The CQP Query Language Tutorial. Technical report, Institut für maschinelle Sprachverarbeitung, Stuttgart, `http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/CQPTutorial/html/`.

Hausmann, Franz Josef (2004). Was sind eigentlich Kollokationen? In Karin Steyer (ed.), *Wortverbindungen – mehr oder weniger fest*, Institut für Deutsche Sprache [= Jahrbuch 2003], 309–334, Berlin: DeGruyter.

Heid, Ulrich (1998/1999 (2000)). A Linguistic Bootstrapping Approach to the Extraction of Term Candidates from German Text. *Terminology* 5(2):161–181.

Heid, Ulrich, Stefan Evert, Bettina Säuberlich, Esther Debus-Gregor, and Werner Scholze-Stubenrecht (2004). Supporting Corpus-Based Dictionary Updating. In *Proceedings of the XIth EURALEX International Congress*, volume I, 255–264, Lorient: UBS.

Heid, Ulrich and Marion Weller (2008). Tools for Collocation Extraction: Preferences for Active vs. Passive. In *Proceedings of LREC-2008*, Marrakesh, Morocco: Linguistic Resources and Evaluation Conference.

Kermes, Hannah (2003). *Off-line (and On-line) Text Analysis for Computational Lexicography*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, arbeitspapiere des Instituts für maschinelle Sprachverarbeitung (AIMS), volume 9, number 3.

Kilgarriff, Adam, Pavel Rychlý, Pavel Smrz, and David Tugwell (2004). The Sketch Engine. In *Proceedings of EURALEX-2004*, 105–111, Lorient, France.

Picht, Heribert (1987). Fachsprachliche Phraseologie – die terminologische Funktion von Verben. In Hans Czap and Christian Galinski (eds.), *Terminology and Knowledge Engineering*, 21–34, Frankfurt: indeks.

Schiehlen, Michael (2003). Combining Deep and Shallow Approaches in Parsing German. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, 112–119, Sapporo, Japan.

Schmid, Helmut (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, 44–49, Manchester, UK.

Schmid, Helmut, Arne Fitschen, and Ulrich Heid (2004). SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection. In *Proceedings of LREC-2004*, Lisboa.

Ulrich Heid, Rufus H. Gouws (2006). A Model for a Multifunctional Electronic Dictionary of Collocations. In *Proceedings of the XIIth Euralex International Congress*, 979–988, Torino.

Zinsmeister, Heike and Ulrich Heid (2003). Significant Triples: Adjective+Noun+Verb Combinations. In *Proceedings of Complex 2003*, Budapest.

A tool for corpus analysis using partial disambiguation and bootstrapping of the lexicon

Kurt Eberle, Ulrich Heid, Manuel Kountz and Kerstin Eckart

Abstract. We describe a tool for the syntactico-semantic analysis of corpus text, which is based on dependency grammar parsing and used for data extraction. It allows for the underspecified representation of structural ambiguities as well as of different lexical semantic readings of linguistic objects.
If required to verify or falsify a given hypothesis, the system partially disambiguates an analysis; besides this, it supports the interactive identification of conditions on such disambiguations. The system has been implemented and is used to examine the sortal properties of German deverbal nominalizations with *-ung*.

1 Introduction

Examinations of corpus text with regard to particular phenomena depend on how precisely such phenomena can be described in a query, as well as on how much the distinctive criteria used in the query play a role during the analysis of corpus text.

Description and analysis of phenomena in turn depend on the degree to which the linguistic examinations targeting these phenomena state usable hypotheses.

Analysis and extraction must consider this dynamics to be ergonomic. For this reason, we represent corpus sentences together with their syntactico-semantic analyses and with lists of parameters relevant to the disambiguation process, thus aiming at flexibility in two ways. Firstly, the depth of individual analyses (i.e. the degree of abstraction reached by the analyses, cf. Kay et al. 1994) may vary. Secondly, we provide a search mode allowing to search for all types of data (textual form of the sentence, analysis, parameter lists) in arbitrary combinations and with varying depth of analysis; thus additional knowledge needed to improve existing theories can interactively be extracted, grouped, examined, and used for deeper analyses.

1.1 Project background

The tool is being developed in the context of work on sortal disambiguation of German nominalizations[1] with the suffix *-ung*[2]: depending on the semantics of the un-

[1] nominalization

derlying verbs, *-ung* nominals can refer to events (e), states (s) or to various kinds of objects (o: concrete, abstract, ...objects; facts), cf. Roßdeutscher et al. (2007). Thus, *Messung* (measuring/measurement) can refer to a process (e) or the resulting object ("data", o); *Teilung* (division) can be – besides the process – the state of being divided, and *Abdeckung* (cover(ing)) has all three readings (cf. Ehrich and Rapp 2000).

The context partners of an *-ung* nominal allow to determine which particular sortal reading is present in a given sentence. *Modifiers* (attributive adjectives, relative clauses, genitive attributes, postnominal PPs etc.) and *selectors* (verbs taking a nominalization as their complement, subject, or adjunct) are often relevant for disambiguation. However, not all modifiers and selectors can be used for disambiguation. We call *indicators* those modifiers and selectors which influence sortal disambiguation.

Spranger and Heid (2007) describe a theoretical framework for the context-based disambiguation of *-ung* nominals. This paper develops their approach further and provides a practical implementation.

1.2 Aims of the tool

Indicators must be identified in order to make predictions about individual readings. E.g. the event reading of *Messung* (i.e. measuring) is preferred if the noun is the object of one of the selectors *durchführen, vornehmen, beginnen, in Auftrag geben* etc. A combination of *Messung* and *liegen bei* + VALUE indicates an object reading.

The task of identifying indicators is done interactively, by inspection of suitable instances from corpora and subsequent generalization.

Conversely, it must be possible to make available to the system new knowledge that identifies particular modifiers and selectors as indicators of a specific type and to take this knowledge into account during subsequent steps of extraction and evaluation.

Our tool supports both processes, thus implementing an instance of the acquisition spiral (i.e. bootstrapping) widely used in corpus linguistics: search – evaluation of results – refined search – better results. The method we implement is special insofar as it applies to the tool and the analyses proper. The system is improved during cycles of analysis and test (of lexical theories) by integrating new or adapted lexical knowledge, which is used during further tests may specify existing analyses further.

Sentences usually contain many structural ambiguities. Most of them have no impact on the investigation of a particular phenomenon and should not lead to an

[2] As part of project B3, *Disambiguation of nominalizations during data extraction from corpus text*, in the SFB-732, *Incremental Specification in Context* (Stuttgart).

increased number of sentence analyses. But some ambiguities do have an impact on the phenomenon in question; it is necessary to consider such ambiguities and their disambiguations appropriately. This affects the reference of indicator candidates in the case of *-ung* nominals.

Thus, sentence representations should be explicitly underspecified; besides this, the system must distinguish ambiguities which have an impact on a given extraction task (i.e. which affect indicators for the sortal disambiguation of *-ung* nominals in our case) from those which do not. It must be possible to disambiguate relevant cases of ambiguity, but with minimal consequences on the rest of the representation.

An example where a structural ambiguity influences the reading of a nominalization is shown in (1). The PP *in Gomel* can be an adjunct to (i) *Messung* or to (ii) *des Radiologischen Instituts*. In case (i) *in Gomel* can fulfill the conditions for an e reading indicator for *Messung*[3] (compare *Messungen [in Innenräumen], Messungen [am Straßenrand], ...*). In case (ii) *in Gomel* is part of the genitive phrase *[des Radiologischen Instituts [in Gomel $_{PP}$] $_{NP}$]* and thus is not available as an (e) indicator for *Messung*.

(1) *Messungen des Radiologischen Instituts in Gomel haben gezeigt, dass die vor Ort erzeugte Milch hoch mit Strontium 90 belastet ist*
Measurements/-ings by the radiological institute in Gomel have shown that the milk produced nearby is massively contaminated with strontium 90 (Stuttgarter Zeitung, 1993)

The system must identify (1) as a sortal disambiguation problem, be able to produce a suitable underspecified representation for it, and disambiguate it appropriately. It must nevertheless be flexible enough to be used for other extraction tasks.

1.3 Comparable Work

It could be argued that parsing and corpus query, which are the main tasks of the system we propose, could as well be done by existing tools, without devising a new approach. Instead of a syntactico-semantic analysis, treebanks could be used (e.g. Negra, TiGer, Tüba, for German). The main problem with existing treebanks is that structural ambiguities are resolved during annotation; besides this, we want to use arbitrary, not necessarily annotated corpora.

[3] Spatially localizing adjuncts occurring in an NP headed by *Messung* typically indicate an e reading for *Messung*. An o reading seems possible, but forced. Temporal localizations (*die Messungen [im Januar]$_{TEMP}$*, but not *die Messungen [von/vom Januar]*) appear to allow only an e reading.

Existing corpus query tools allow to query syntactically annotated corpora to some extent (compare CQP, Evert 2005, `tgrep`, etc.). However, most corpora – except treebanks – are only annotated at the level of word forms (lemmas and parts of speech) or chunks. More complex constructions, especially extraposed PPs, relative clauses, passivization etc. can only be retrieved with low recall and/or precision, if at all; or the formulation of queries becomes cumbersome.

Any formal grammar could, in principle, be used as a starting point to assign underspecified representations to (sets of) corresponding syntactic analyses of corpus sentences. Our approach uses *slot grammar* (McCord 1991) as implemented and used in the research version of the MT tool *translate*[4], because its dependency-style analyses are especially apt to be mapped into underspecified representations; besides this, appropriate procedures are already available and can easily be adapted to the specific needs of corpus investigation respectively (cf. Eberle 2002).

Underspecified representations of ambiguities are, in our view, more suitable for corpus search than packed parse forests (as they are produced e.g. by the XLE implementation of LFG, cf. Maxwell and Kaplan 1991), because they are much more compact, and searching is more economical.

We are also assessing the possibility of using a system based on (Schiehlen 2003) as an alternative to the adapted *translate* analysis.

2 Architecture

2.1 Requirements

Firstly, the system must provide a flexible interface to the lexicon, in order to allow us to use variably elaborate descriptions of the meaning of a lexeme (e.g. a modifier as an indicator of a particular sortal reading) *without* changing the processing components of the system.

Secondly, the system must assign underspecified representations to corpus sentences, and it must be able to search these; and thirdly, it must be able to carry out partial disambiguations. Partial, here, means firstly to only reduce lexical ambiguity instead of resolving it, if needed; secondly to make relations among words more precise, instead of fixing them; and thirdly to disambiguate only a substructure, but not necessarily a whole sentence.

The system employs *Flat underspecified Discourse Representation Theory* (FUDRT, Eberle 1997, 2004) for this task; FUDRT is an extension of UDRT (Reyle 1993) which also handles lexical and functional ambiguities. [5]

[4] `http://lingenio.de/English/Products/translation-software.htm`

2.2 Lexical representation

Lexemes are mapped into *labeled* functional terms that range over semantic representations and are evaluated (partly) depending on conditions connected to them. Evaluation may relate either to solving an ambiguity, or to assigning a more fine-grained representation. A label identifies a subrepresentation of the complete sentence representation; it is decorated by the *distinguished discourse referent(s)* (DRFs) introduced by this subrepresentation.

For example, the representation $l_{x@eso}$: $\underline{\text{absperrung}}(\text{x})$ is assigned to the word *Absperrung* (blocking/fence); underlining marks a term (here, dependent on *x*) as functional. *x* is also the distinguished DRF, being an element of the semantic class eso (where eso stands for e: event, s: state *or* o: object).

$\underline{\text{absperrung}}$ can be evaluated e.g. along the following functional specification:

$$\underline{\text{absperrung}}(\text{ref}) \Rightarrow \text{nsem\_l}$$

$$\underline{\text{absperrung}}(\text{e @ event}) : \text{l}_{e@event}\text{:} \ \boxed{\begin{array}{l} \text{e (s, o)} \\ \\ \text{absperren(e)} \\ \text{abgesperrt(s)} \\ \text{meets(e,s)} \\ \text{absperrung(s,o)} \end{array}}$$

$$\underline{\text{absperrung}}(\text{s @ state}) : \text{l}_{s@state}\text{:} \ \boxed{\begin{array}{l} \text{s (e, o)} \\ \\ \text{abgesperrt(s)} \\ \text{absperren(e)} \\ \text{meets(e,s)} \\ \text{absperrung(s,o)} \end{array}}$$

$$\underline{\text{absperrung}}(\text{o @ object}) : \text{l}_{x@object}\text{:} \ \boxed{\begin{array}{l} \text{x} \\ \\ \text{absperrung(o)} \end{array}}$$

That is, $\underline{\text{absperrung}}$ is a mapping from DRFs into labeled semantic representations of type noun semantics (nsem_l).[6] It depends on the argument *x* being an event e, a state s or an object o, whether the result is a DRS which introduces and describes a DRF for an event, a state or an object (*e*, *s* or *o*) and makes it available "to the outside" (for compositional semantic construction).

In this exemplary interpretation of *Absperrung*, further conditions are introduced in the first two meanings. They describe consequences as can be drawn from the existence of the introduced distinguished DRF: If there is an event *e* of *blocking a road*,

[5] Its development traces back to implementations of UDRT (Reyle 1993) within the SFB 340 at the University of Stuttgart; it is an attempt to have underspecified representations available for broad language fragments. UDRT is the first of a number of underspecification formalisms that have been developed since the early nineties.

[6] More precisely, a function from individuals into the meaning of a noun representation, which is typically a function from individuals into truth values or functions form situations into truth values, respectively.

then there is also a pertaining consequent state s and some kind of object o keeping up this state; this is similar for the state s of the second meaning (there must have been an event causing s etc.). For the object o, no such interrelations can be deduced. Deduced DRFs are not automatically available as antecedents for anaphoric relations; they are written (in the specific DRS dialect illustrated by these examples) in parentheses in the designation of the DRS universe (i.e. $e(s,o)$, $s(e,o)$). [7]

The internal structure of evaluations of the functional description is not taken into account when the sentence representation is constructed, but only the sortal properties of the distinguished DRF. This means in particular that the representations resulting from such evaluations can be replaced by representations with a (yet) finer structure, or by other representations derived e.g. using common derivation rules. This leaves a certain freedom for structuring the lexicon; it also means that the system can be used to test different interpretations of given lexical material without changes to the components of the system. Only modifications which influence the sortal interpretation do have an impact on the construction of representations.[8]

2.3 Compositional sentence semantics

During the process of compositionally constructing the sentence representation, complements and adjuncts are not applied to their arguments in the sense of a beta conversion in the lambda calculus; rather, they are recorded in the *functor set* of the base representation (of the noun, verb etc.). A set of constraints defines how far the possibilities to relate these partial representations with each other are restricted beyond the requirements inherent to the representations (i.e. given by syntactic and logical type constraints). This set of constraints may be empty.

The following (artificial) example gives a compact illustration of the shape of such FUDRT representations, and of the kinds of conclusions which can be drawn given certain sortal properties. We skip all details of the representation which are irrelevant to the topic; in particular we skip representing the tense information. (L_i are representations, and l_i the pertaining labels):

(2) *LKA-Mann Peterson plante stundenweise Absperrungen mit verschiedenen Materialien.*

(i) Peterson planned hour-long blockings using all kinds of materials.

(ii) Repeatedly for hours, Peterson planned barriers consisting of all kinds of materials.

[7] The functional modelling allows for various interpretations in DRT or other representation languages. In particular, there might be interpretations for partial sortal knowledge, for the non-object reading (`es`) for example.

[8] If logical inferences beyond sort subsumption within a sortal hierarchy are used, different contents of evaluations can have an influence. This case is explicitly excluded within our project.

$$(2_{rep})\quad \mathrm{l0}_e{:}\boxed{\begin{array}{l} \text{e} \\ \\ \text{planen(e)} \\ \text{agent(e) = p} \\ \text{theme(e) = X}\end{array}} \left\{ \begin{array}{l} \mathrm{l1}_p{:}\boxed{\begin{array}{l}\text{p}\\ \text{peterson(p)}\\ \underline{\text{lka-mann}}\text{(p)}\end{array}}, \mathrm{l2}_{E'}{:}\underline{\text{stundenweise}}(\text{L'}2_{e'@event}), \mathrm{l3}_{X@eso}{:}\underline{\text{absperrung}}^*(\text{X}), \\ \\ \mathrm{l4}_Y{:}\underline{\text{mit}}(\mathrm{l5}_Y{:}\underline{\text{verschieden}}(\underline{\text{material}}^*(\text{Y})), \text{L}'_{5_{z@object\vee e\_cog}}) \end{array}\right\}$$

Four functors are given in the representation; their applicability to L_0 or one another is only restricted by sortal requirements on their respective arguments and by information from the syntax-semantics interface: *LKA-Mann Peterson* is the agent and *Absperrungen* is the theme of e, *stundenweise* modifies a representation with an event as the distinguished DRF (e'), and, as a temporal quantification, outputs a set of events (E'). The *mit*-PP modifies either an object or a creation event (z).

2.4 Partial disambiguation

Three specifications of applicability constraints are possible with this example, all of which lead to different sortal interpretations of the string *Absperrungen*. Other possibilities are excluded because of sortal clashes. (a) *stundenweise* can modify *Absperrungen*; because e' and X will be identified, we can conclude that the several *Absperrungen* are events (`@event`) in this case (*stundenweise Absperrungen*). (b) *mit verschiedenen Materialien* can modify *Absperrungen* as well, which are then (planned) objects (`@object`). Due to their sortal requirements, these two modifiers cannot both refer to *Absperrungen* simultaneously (an event of *blocking* is not a cognitive event). In this case, (c), *Absperrungen* remains sortally underspecified.

The analysis tool extracts sentences from the corpus which contain *-ung* nominals and assigns to them underspecified representations; these can be disambiguated as described above. Disambiguation is considered only as far as it is relevant for the given task; corresponding constraints are added, but no functor applications interpreting them are carried out. In the example case, there are only constraints that are relevant, because all of them influence the sortal interpretation of the *-ung* nominal: In oder to obtain reading (a), the set of constraints in (2_{rep}) is extended by adding `first(l3,l2)`, which says that the representation L_2 must be related to L_3 before L_3 is related to the base representation L_0 or to a DRS which results from L_0 via applying functors to it. Relating L_2 to L_3 means that L_2 is pushed to the functor set of L_3; logically, this is tantamount to raising the type of $L3$. In (b) the set of constraints is analogously extended by `first(l3,l4)`; (c) only requires that none of these two type raisings take place, but it is not determined whether L_2 should have scope over L_4 (when applied to the verb) or vice versa.

Parameters which are possibly relevant to the disambiguation process but not treated so far are easily extracted from the underspecified sentence analyses (i.e.

modifiers and selectors which can – but need not – refer to an *-ung* nominal and which are not yet classified).

The results are fed back into the system as extensions of the respective lexicon entries, in terms of functional descriptions as shown above. Thus, the sortal interpretation of a given sentence can change dynamically, and the impact of modifications of the lexicon can be tested.

The system we describe is implemented as an adaptation of the research version of the MT tool *translate*. It computes underspecified analyses both for single sentences and for sentence lists. It does partial disambiguation of analyses as described, and provides several additional test and inspection routines, in particular for parameter extraction. It also supports dynamic extension of the dictionaries. [9]

3 Sample results and possible applications

The system has been applied to a small corpus of 363 sentences containing instances of *Messung*, in order to obtain a basic set of parameters relevant to the sortal disambiguation process. Based on this set, 6000 analyses were obtained for a fictional text and 30000 analyses for a corpus of randomly selected newspaper texts; these are currently examined for further indicators and regularities. [10]

3.1 Extraction of parameters

The tool can produce output representations with varying granularity: With and without reference to the passage in the text or corpus, with and without giving particular information about the concrete circumstances of use (morphosyntactic information, occurrence of words, or semantic type), see (3,4):

(3) Selectors:
[feststell,vprep(bei),xprep(bei),xprep(nach)]
[gelt,iobj(p([bei|dat])),subj(n),xprep(von)]
...

Modifiers:
[nprep(in),frühjahr]
[xprep(in),grundwasser]
[nprep(in),monat]
...
[nadj,monatelang]
[xmod,stundenweise]
...

[9] The system is written in C++/Prolog and runs in Windows environments. Because it uses commercial software stemming from IBM's *Logic based Machine Translation* project (LMT) and has been continuously exposed to the market, improved and extended for over 15 years, the coverage of grammar and dictionary is remarkably large. Since 1997 FUDRSs are used in LMT for semantic representation.

[10] Fiction: Flaubert, German translation of *Mme Bovary* (Project Gutenberg); Newspaper corpus: Corpus of the project *Wortschatz*, University of Leipzig.

(4)

```
[n(feststell,mtv(ind:dcl:nwh,tf(past,0,_),a), [cogv]), [xprep(bei), [166]], [xprep(nach), [293]]]
[n(feststell,mtv(ind:dcl:nwh,tf(past,0,_),p), [cogv]), [vprep(bei), [40,297]]]
...
[nprep(in), [n('frühjahr',det(def,sg), [season0]), [40]]]
[xprep(in), [n(grundwasser,det(def,sg), [liq]), [256]]]
[nprep(in), [n(monat,det(def,pl), [intervall0,timeas]), [257]]]
...
```

The output given as (3) reports that *-ung* nominals occurred as the internal arguments of various PP verb modifiers (PPs with *bei* and *nach* below *feststellen*, and PPs with *bei* and *von* below *gelten*). In sentences with *gelten*, an *-ung* nominal occurred at least once as the subject (`subj(n)`). *-ung* nominals where modified by PPs with *in*, where *Frühjahr, Grundwasser, Monat* occurred as head nouns of the internal NP, and by adjectives like *monatelang* and *stundenweise*.

Relations for selection and modification are specified as detailed as is justified by syntactic constraints and lexical information on valency: Thus, a PP with *bei* occurred at least once as an indirect object of *gelten*, at least once as an adjunct to *feststellen*, and at least once such that the PP could modify a VP with *feststellen* (but not necessarily). The situation was similar for (possible) modifiers to the *-ung nominal* with regard to the distinction between `nprep` and `xprep`, `nadj` and `xmod`.

An excerpt from (3) is given as (4), where additionally the morphosyntactic characterization, semantic types of selectors and arguments and the respective corpus positions are given. Individual instances can be identified as different subcases beyond the view in (3), due to the refined classification: E.g. *feststellen* occurs in active voice in sentences 166 and 293, in passive voice in sentences 40 and 297 (both with a simple past tense form, preterite, in matrix position, `ind`, of a declarative sentence, `dcl`, without any wh-element, `nwh`). Subcategorized functions always relate to deep structure in these presentations.

Indications on the semantic typing should help to identify regularities, thus allowing to cluster significant groups of occurrences and to define indicators which are as generic as possible for the disambiguation of *-ung* nominals, e.g. [`xprep(in), intervall0`] and [`xprep(in), mass`] as indicators for the *e* reading of *-ung* nominals like *Messung*.

Such properties can only be considered during analyses, when they previously have been included in the lexicon. This means in particular that in sentences like (2) disambiguation with both `xmod` and `xprep` modifying the *-ung* nominal (*stundenweise Absperrungen in verschiedenen Größen*) is only prohibited if the modifiers are recognized as instances of conflicting indicators.

Currently, we use the sentence analyses available for the three corpora mentioned above to test working hypotheses on disambiguation. We assume, for example, with Roßdeutscher et al. (2007) that the sortal characteristics of individual *-ung* nominals follow from the structure of their respective morphological derivations.

Potential selectors and modifiers of different classes of -*ung* nominals are extracted as described above, their usability as indicators for sortal distinctions is assessed and compared to previously gathered data; the results are entered into the lexicon. This additional lexical knowledge further restricts the options for disambiguation with regard to sortal readings of -*ung* nominals. Thus, the calculation of such options for the available sentence analyses provides a basis for a quantitative review of hypotheses.

3.2 Sortal paradoxes

Quantitative and qualitative estimations regarding the problem of (supposedly) conflicting sortal requirements (called *sortal paradox* e.g. by Brandtner (2008)) can be done in a similar way:

(5) *Die Übersetzung dieses Werks konnte bereits 1990 abgeschlossen werden (e) und als erster Band des Gesamtprojekts erscheinen (o).*
The translation of this work could be terminated already in 1990 and published as first volume of the whole project. (Google cited by Brandtner (2008))

Given that all needed information is contained in the lexicon, similar sentences may easily be found by applying the disambiguation procedure described above to sentence analyses, and then having the system identify sentences which exhibit or may exhibit a sortal conflict as in (5) and (2) respectively.

3.3 Fact readings

A topic which can be seen as a special case of sortal paradoxes is the investigation of -*ung* nominals which, due to the sentence structure, seem to receive an interpretation as a shortened version of a description of a fact – contrary to the expectation of e, s or o. This is the case if they occur in the position of a clausal complement or if the nominalization is – mediated by the verb – logically related to a sentential complement (cf. 6_{res} for examples).

Investigations like this can be done easily using sentence templates. As it is crucial to also retrieve sentences which only *can*, but need not have the readings in question, we exploit underspecified representations of variable depth also for formulating queries.

Using the representation in 6_{rep} as a query for representations of (parts of) sentences yields examples where an -*ung* nominal is the subject, and which contain a non-fact-indicator possibly modifying the nominalization, and where a clausal representation occurs (possibly constructed from a *dass* clause) which may (but not necessarily must) modify the (representation of) the VP.

(6$_{rep}$)

$$\begin{array}{ll} & \{\ \mathrm{l1}_{u@eso}\text{:ung\_deriv(l'}1_{e'@event}\text{:L'}_1\text{)},\quad \mathrm{l2}_{u'}\text{:}\underline{\text{mod}}\text{(L'}2_{x@eso}\text{)},\quad \mathrm{l3}_{e''}\text{:}\underline{\text{fin}}\text{(e'')}\ \} \\ & \{\diamond\ \text{first}(\mathrm{l}_1,\mathrm{l}_2)\ \} \\ \mathrm{l0}_e\text{:L}_0\ \&\ \boxed{\begin{array}{l}\text{e}\\ \text{subj(e,u)}\end{array}} & \end{array}$$

Here, $\texttt{ung\_deriv(l'}1_{e'@event}\texttt{:L'}_1\texttt{)}$ says that the *-ung* nominal be derived from a root whose semantic representation introduces an event e', thus forcing the nominalization to have an *e* reading, according to Roßdeutscher et al. (2007). $\underline{\texttt{mod}}\texttt{(L'}2_{x@eso}\texttt{)}$ indicates that the modifier refers to arguments which must be *eso*; $\underline{\texttt{fin}}\texttt{(e'')}$ tells us that the functor in question must be clausal (finite) and must describe an event e''; $\diamond\texttt{first(l}_1\texttt{,l}_2\texttt{)}$ indicates that the representation labeled l_2, i.e. L_2, can be applied to the representation labeled l_1, i.e. to L_1.

(6$_{res}$) is an excerpt from results for the newspaper corpus we analyzed:

(6$_{res}$) *Die Verschärfung der Korruptionsgesetze im Jahr 1997 hat bewirkt, dass selbst das so genannte "Anfüttern" von Beamten mit Geschenken ohne Gegenleistung als strafbar gilt.*
Und weiter heißt es in der vom braunen Zeitgeist geprägten Broschüre: "Dass die nationalsozialistische Gedankenwelt Allgemeingut der ganzen Kolonie geworden ist, zeigt die Beteiligung aller Volksgenossen am Winterhilfswerk und an gemeinsamen Eintopfessen."
Der Arbeitsalltag der Hausdame ist bestimmt von der gelebten Überzeugung, wonach eine Frau in ihrem Fach entweder "mit einem Mann oder einem Hotel verheiratet ist".

Note that the first and second sentence each contains an *e* indicator which can refer to the respective nominalization but need not: In the first sentence, *im Jahr 1997* can refer to *Verschärfung* (aggravation), and *am Winterhilfswerk* (winter relief organization) to *Beteiligung* (participation) in the second sentence. In the third sentence, the finite modifier is not constructed from a clause beginning with *dass*, thus the more abstract type given in the query becomes effective. Besides this, the modifier does not necessarily refer to the *-ung* nominal (*Überzeugung*; which is subject of the deep structure of the analysis), but may also refer to the sentence VP.

Results for queries like this are used to further subclassify the verbs occuring in these sentences, first of all, to clarify which selection restrictions (and for which reasons) appear to be appropriate for subjects of verbs like *bewirken* (cause), *zeigen* (show), *bestimmen* (affect) or *bestätigen* (confirm), *ergeben* (result), *einhergehen mit* (come along with) etc. when used as relating to facts.

The analysis tool flexibly shows the consequences of the various modelling options without any need to change its processing components. If, for example, in the context of the above discussion of verbs which present relations to facts, we require for a verb or a class of verbs (or for a certain way to use it or them) that the subject be not of class `event`, then the tool will mark sentences as in (6$_{res}$) as possibly irregular, because the (dominant) readings of these sentences lead to sortal clashes. (Implicitly, this presents to the inspector the task to investigate the clashes, to evaluate how serious the respective violations of appropriateness are and how easily a potential reader or hearer could be ready for cooperative "repair" in a Gricean sense).

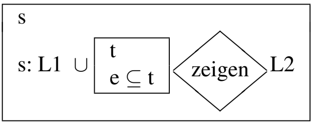
If, on the other hand, `event` is admissible by the selectional restrictions for the subject, because the respective theory which is tested assumes that an event is a natural cause and starting point of the situation described in the finite complement, then these sentences are not recognized as being special.

Third option: Should the considered theory assume for such cases specific selectional restrictions but suggest interpretations which "repair" the violations in a regular manner, then the functional approach to modelling the semantic contributions of words makes it easy to add corresponding rules to the lexicon.

If a fact is expected but an event is introduced, a regular possibility of "repairing" is, for example, to presuppose the existence of the introduced event in the time space of the text representation and to interpret the expected fact as this presupposition. (6_{lex}) shows a part of the lexicon entry for *zeigen* where a corresponding use (as in (6_{res}) second sentence) is modeled:

(6_{lex}) $\underline{\text{zeigen}}(\text{npsem,fin}) \Rightarrow \text{vpsem}$

$\underline{\text{zeigen}}(\text{L1}_{e@event}, \text{L2}_{e'@event}) := \text{l3}_{s@event}:$



It is a question of modelling in how far such lexical rules can be called "repair" (and thus be output by the system as such), or in how far they can be seen as regular interpretation rules which are unmarked in a Gricean sense.

As opposed to the modelling given in (6_{lex}), there is also the option of binding the construction of a fact rather to the subject NP than to the verb. From the perspective of formal logic, one will prefer that way of modelling which is more general: If the choice of extension depends more on the particular kind of subject description than on the type of predication, then the subject NP is a better place for this. This seems not to be the case here, thus (6_{lex}) appears to be appropriate.

4 Conclusions and next steps

We presented a tool for the syntactico-semantic analysis of corpus text, illustrating the approach by means of possible applications and with selected phenomena which are important to the sortal disambiguation of German *-ung* nominalizations. The tool carries out a deep analysis (based on dependency grammar), and produces underspecified representations of ambiguities (using FUDRT). It has a modular and thus flexible architecture: it allows to work with lexical information of variable granularity, and thus it can accommodate different approaches to modelling a specific phenomenon (see the discussion in section 3.3) and allows to compare these with regard to their impact on processing, disambiguation, and options for representing the results.

Current work covers a detailed analysis of cases of sortal paradoxes (see 3.2) and the conditions of fact readings of *-ung* nominals (see 3.3); we analyze large amounts of corpus text for this task. In addition, more phenomena in the field of disambiguation of *-ung* nominals, and other data extraction tasks are planned to be investigated (e.g. the subcategorization behaviour and nominalization of German particle and prefix verbs, extraction of collocations etc.). The results are to be assessed for evaluating and optimizing the tool. A database is being implemented which allows to compare the results of this tool to those from other tools for syntactic analysis (e.g. Schiehlen 2003). In this database, corpus data, analyses and metadata are brought together, ranging over different levels of linguistic description, different tools, and several approaches differing in their theoretical foundation.

References

Brandtner, Regine (2008). Meaning Transfer and the Compositional Semantics of Nominalizations. In Florian Schäfer (ed.), *SinSpeC. Working Papers of the SFB 732*.

Eberle, Kurt (1997). Flat Underspecified Representation and its Meaning for a Fragment of German. Arbeitspapiere des Sonderforschungsbereichs 340 120, Universität Stuttgart, Stuttgart.

Eberle, Kurt (2002). Tense and Aspect Information in a FUDR-based German French Machine Translation System. In Hans Kamp and Uwe Reyle (eds.), *How we say WHEN it happens. Contributions to the Theory of Temporal Reference in Natural Language*, 97–148, Tübingen: Niemeyer, Linguistische Arbeiten 455.

Eberle, Kurt (2004). Flat Underspecified Representation and its Meaning for a Fragment of German. Habilitationsschrift, Universität Stuttgart, Stuttgart.

Ehrich, Veronika and Irene Rapp (2000). Sortale Bedeutung und Argumentstruktur: *-ung*-Nominalisierungen im Deutschen. *Zeitschrift für Sprachwissenschaft* 19(2):245–303.

Evert, Stefan (2005). The CQP Query Language Tutorial. Technical report, Institut für maschinelle Sprachverarbeitung, Stuttgart.

Kay, Martin, Jean Mark Gawron, and Peter Norvig (1994). *VERBMOBIL: A Translation System for Face-to-Face Dialog*. Stanford: CSLI Publications.

Maxwell, John T., III and Ronald M. Kaplan (1991). A Method for Disjunctive Constraint Satisfaction. In Masaru Tomita (ed.), *Current Issues in Parsing Technology*, 173–190, Dordrecht: Kluwer.

McCord, Michael (1991). The Slot Grammar System. In Jürgen Wedekind and Christian Rohrer (eds.), *Unification in Grammar*, MIT-Press.

Reyle, Uwe (1993). Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics* 10(2):123–179.

Roßdeutscher, Antje, Hans Kamp, Uwe Reyle, and Torgrim Solstad (2007). Lexical and Supra-Lexical Underspecification Rooted in a DM-Based Theory of Word Formation, ms., SFB-732 Jahrestagung.

Schiehlen, Michael (2003). A Cascaded Finite-State Parser for German. In *Meeting of the European Chapter of the Association of Computational Linguistics*.

Spranger, Kristina and Ulrich Heid (2007). Applying Constraints derived from the Context in the process of Incremental Sortal Specification of German -ung-Nominalizations. In *Proceedings of the 4th Int. Workshop on Constraints & Language Processing, CSLP*.

Rapid construction of explicative dictionaries using hybrid machine translation

Kurt Eberle and Reinhard Rapp

Abstract. We describe an approach that largely automatizes the construction and extension of MT dictionaries. These dictionaries include selectional constraints allowing informed decisions with regard to the selection of translation alternatives. Using an MT system with a rule based core, bilingual corpora are analyzed and syntacto-semantic conditions are extracted that are useful for the selection of translation alternatives. For illustrative purposes, using the generation module of the MT system the extracted rules can be transformed to explicative examples of usage. In contrast, translation knowledge used by current statistical MT systems is usually not well suited to be prepared for human inspection. Of particular importance for our approach is the underspecified semantic representation which is the level of analysis to be used for the formulation of all translation relations. The approach is currently being investigated using components of Lingenio's MT system *translate*.

1 Introduction

There is a great variety of MT dictionaries, with differing contents for a range of intended purposes.

Traditional dictionaries have the user in mind and typically inform him about categories such as the morphological class, subcategorization, and possibly about the semantic typing of a lemma, its homonyms, or its homonym readings. In the case of ambiguous words with several possible translations, selection criteria are usually provided. These can be selectional restrictions of complements or information on typical contexts. Often these constraints and informations give only examples in the sense of "as used in the following" or "as in the following situation". Such examples are often not precise enough to provide clear instructions about how to choose from the alternatives in specific contexts.

In contrast, MT dictionaries are designed to exactly do this. They are used by the MT system in such a way that for arbitrary contexts a single alternative is chosen, or, alternatively, a ranking of all possibilities is computed. The solution the system comes up with can only be understood by the user if the lexical knowledge is prepared in a human-readable form, i.e. with information types as known from traditional dictionaries.

In a lexicon as used for statistical machine translation the relevant selectional contexts are usually n-grams or typed n-grams, i.e. a given word's neighborhood of width *n* (typically $\leq$ 5–10 words). Alternatively, generalizations of words to multiword units can be used. In all cases, word forms can be replaced by base forms or (less commonly) by semantic types. For each n-gram a probability is provided which quantifies the selectional preference of a particular translation in the respective context (cf. Manning and Schütze 1999: chapter 13 for the foundations; also Brown et al. 1992; Och and Ney 2002). Typically, such a lexicon provides a very large number of such descriptions for each word, lemma, or meaning. These descriptions are unstructured lists, and it is non-trivial to generate human readable dictionaries from them.

In contrast, dictionaries in a modular rule based MT system can relate to the syntactic or semantic representation level the system uses for transfer and can define selectional constraints as structural properties of such representations, thereby summarizing a multitude of individual cases by one description (cf. Eberle 2001; Emele et al. 2000).

In comparison to n-gram knowledge the advantages are the smaller number of cases and the fact that the constraints stand for coherent linguistic units which are familiar to the user rather than word lists that often disregard structural boundaries. An advantage over traditional dictionaries is the verifiability by context. The main disadvantage is that for lexicographers it is very time consuming to spell out the contextual constraints in sufficient detail. Consequently, as tests have shown, even sophisticated rule based MT systems show gaps with regard to having all types of uses of a word exhaustively or adequately modeled. We therefore discuss in the following how this kind of operational knowledge can be automatically extracted from corpora.[1]

2 Aims and procedure

We pursue a bootstrapping approach where the MT system's analysis components are used to analyse aligned sentence pairs of the source and the target language. Hereby the information from the existing translation lexicon is exploited to mutually relate words and structures of the analysed sentences, thereby discovering new translation knowledge and feeding it into the system.

These are the basic steps of the suggested procedure: Construction or adaptation of components for the analysis of the languages to be considered; compilation of a bilingual base vocabulary; extension of the base vocabulary through the analysis of parallel texts.

[1] We cannot consider the topic of generating sample sentences from corresponding representations but instead refer the reader to http://lingenio.de/Deutsch/Sprachtechnologie/FlexiDict.htm where a corresponding technology using the generation component of the system is described.

2.1 Analysis

In principle there are two different possibilities for the extraction of translation knowledge from sentence pairs.[2]

Using classical statistical procedures, it is possible to compute word correspondences, multiword correspondences, and base form correspondences without requiring analytical knowledge beyond the word or multiword boundary. A common algorithm for doing so is GIZA++ (cf. Och and Ney 2003).

Alternatively, it is also possible to generate syntactic or semantic representations and to compute relationships on the basis of the information from these levels of analysis.

It is a drawback of the classical, uninformed statistical models that the possible correspondences cannot be constrained by linguistic information on categories, or that the language model that may provide corresponding information must first be computed from the raw data in an error prone process. Also the sparse data problem is omnipresent and in many cases prevents to make well-founded choices.

A disadvantage of procedures with classical syntactic and/or semantic analyses is that these, to be applicable to practical problems, presuppose grammars and (monolingual) dictionaries with high coverage. Another disadvantage is that, due to incompleteness and errors with regard to structural and lexical decisions, they sometimes come up with unintuitive or erroneous analyses when processing sentences.

In order to avoid this it has been suggested for purposes like ours to partition a sentence into *chunks*, to analyse the chunks syntactically, but to avoid merging the resulting partial analyses into one full analysis of the sentence (as this last step would be highly error prone, cf. Tsuruoka and Tsujii 2005).

In our view the disadvantage of the chunk analysis is that many constraints between chunks are not optimally taken into account, and that, more generally, the consideration of dependencies decreases with increasing distance. This is analogous to classical statistical models where n-gram lengths need to be small for complexity reasons and as the impact of the sparse data problem increases with larger n-gram sizes.

Detachable prefixes in German are good examples to illustrate how relatively large the distances for translation-relevant dependencies can be. The following examples based on verbs with stem *stell* were extracted from the *Europarl parallel corpus*.[3]

[2] In this context, the term “sentence pair” can be interpreted liberally, for example as describing a pair of sequences of sentences as delivered by algorithms such as Gale and Church (1993).

[3] Collected by Philipp Koehn; version 3 put together by Cameron Shaw Fordyce (CELCT), Josh Schroeder, and Philipp Koehn (University of Edinburgh), cf. `http://www.statmt.org/europarl` (Koehn 2005).

(1) a) *darstellen*: Distance 35 words/punctuation marks
Obwohl uns die Schwierigkeiten seit Jahren bekannt sind, stellt der jetzt zur Diskussion stehende, im Vermittlungsverfahren zwischen Parlament und Rat erzielte Kompromiss möglicherweise den ersten großen, entscheidenden Schritt zur Schaffung eines tatsächlich vereinten Schienenverkehrssystems und zu der so dringend nötigen Liberalisierung dar. (File ep-01-01-31.txt, line 2435)
b) *aufstellen*: Distance 24 words/punctuation marks
Herr Präsident, dieser Bericht stellt zu Recht gerade rechtzeitig vor der April-Tagung des Internationalen Währungsfonds und der Weltbank erneut die Forderung nach einer verbesserten internationalen monetären Zusammenarbeit auf. (File ep-96-04-18.txt, line 170)
c) *einstellen*: Distance 15 words/punctuation marks
Danach stellen wir uns gedanklich schon wieder auf die Haushaltsdebatte in der kommenden Woche im Haushaltsausschuß ein und auf die Abstimmung in Straßburg später im Monat. (File ep-97-10-01.txt, line 723)

If the verb is in verb-second position, the average distance between verb and prefix can be assumed to be around 10 words (the distance also depends on the particular prefix). This means that such phenomena are a major source of error when aligning translations because such long distance dependencies are often not recognized. However, as verbs in general, also prefix verbs and their correct translations are an indispensable part of high quality dictionaries for everyday use.

Syntactic and semantic representations avoid these problems. However, as pointed out previously, they are complex and costly to elaborate and often deliver faulty results.

Parse forests and underspecified syntactic and semantic representations do not resolve (certain) structural (and lexical) ambiguities in cases where the available information is insufficient. This more cautious strategy leads to better founded results, and still leaves the possibility of subsequent (partial) disambiguation and further specification in case additional information (from beyond the local domain or the sentence boundary) becomes available later. From another point of view, this also implies that the analysis component can get along with less information, which is an important advantage as this helps to avoid making poorly justified decisions.

For such reasons, as the outcome of the analysis we suggest underspecified semantic representations. Underspecified representations are more compact than analysis forests, and semantic representations are better suited to abstract away from surface properties than syntactic representations. We therefore consider them more appropriate here.[4]

[4] Nevertheless information from the syntax-semantics interface should be provided.

2.2 Bilingual base lexicon

The base vocabulary serves the purpose to correctly specify non-trivial translation relations. On the one hand, (for both source and target language) it provides monolingual lexical knowledge required for the analysis of source and target sentences. On the other hand, it can be used in analogy to *cognates* (cf. Simard et al. 1992). That is, it provides anchor points with regard to determining the translation relations between sentences, thereby significantly reducing the search space for word- or multiword alignments within sentence pairs.

Rule-based analysis is not possible without having at least a fragment of the basic vocabulary. High frequency words are typically more ambiguous than low frequency words, and their ambiguities tend to be more diverse. Similar considerations apply when comparing verbs to nouns and other parts of speech, i.e. on average the verbs carry more ambiguity.

We propose to manually create a lexicon containing high frequncy words and words that are highly ambiguous, together with their most essential translation relations. This lexicon is chosen to be relatively small, with some preference being given to verbs due to their specific character.

Best suited as cognates (or anchor points) are words which carry little ambiguity and whose translations are also not very ambiguous. Because of these properties, such word pairs can be extracted with high reliability from parallel corpora. By means of analysis they can be efficiently and semi-automatically prepared for import and fed into the translation lexicon.

Bilingual dictionaries for particular subject areas are increasingly available in electronic form. They can be utilized to enhance the coverage of the “cognates” significantly. Again, most of them can be analyzed automatically and be fed into the lexicon together with the necessary morphosyntactic and semantic information.

Experience tells that despite such methods considerable gaps in the basic vocabulary are to be expected. Some more general information is not available in specialized dictionaries, and electronically available parallel corpora are still comparatively small for most language pairs, despite the fact that some progress could be made over the last couple of years. [5]

An important part of the proposed method is to extract associative knowledge from monolingual corpora of both source and target language and to inter-relate it by utilizing homomorphism, thereby significantly expanding the base lexicon. Our algorithm is based on the observation that if two words A and B co-occur more often than expected from chance in a source language corpus, it is likely that their

[5] Version 3 of the Europarl corpus comprises approximately 1.3 million aligned sentences, whereas for example the monolingual Gigaword corpora of the Linguistic Data Consortium contain at least 50 million sentences.

translations τ(A) and τ(B) also co-occur more often than expected in a corpus of the target language. As the respective algorithm is computationally expensive and typically leads to a high error rate, our focus is on formulating structural constraints that help improving the results. Details on the previous work can be found in Rapp (1999) and Rapp and Vide (2007).[6]

An evaluation of results is currently being conducted. It concerns existing lexica of some languages that are still under construction in the research version of the *translate* system (further information on *translate* is given below).

Following these considerations, the compilation of a base lexicon requires the following steps:

a) Manual compilation of a bilingual dictionary for high frequency words that are highly ambiguous.
b) Semi-automatic statistics-based and analysis-controlled expansion of the dictionary. This way high frequency words that are less ambiguous are added together with their translations.
c) Largely automatic analysis-controlled expansion of the coverage using specialized technical dictionaries.
d) Statistics based extension of the lexicon using associative knowledge derived from pairs of monolingual corpora.

For steps b), c), and d) the existing lexical-relational knowledge from the previous step is used to improve the results. Step d) involves recursion of step b) towards less frequent and more ambiguous words.

2.3 Semi-automatic lexicon expansion

Besides automatizing the compilation of the base lexicon, the main focus of our approach is on adding entries related to ambiguous words and to find their correct translations. This is the most expensive activity when building up a lexicon, and also the most error prone. Herefore, we use the base lexicon as a source of knowledge. Let us consider some examples concerning translations of the German verb *einstellen* into English. In the Europarl corpus we find, among others, the following usage samples and translations.

(2) a) *jmd. einstellen – to hire*
Es kann jedoch die Chefs einstellen und feuern.
What it can do is hire and fire the bosses. (File ep-05-05-25.txt, line 165)

[6] A related approach with a focus on *paraphrasal* knowledge is Callison-Burch et al. (2006).

b) *etw. einstellen – to adjust*
Die Automaten, um die es ja hauptsächlich geht, können entsprechend eingestellt werden.
The vending machines - which are the main issue - can be adjusted. (File ep-06-03-14.txt, line 3991)
c) *sich auf etw. einstellen – to adapt*
Große Task-forces werden eingesetzt, die Industrie stellt sich darauf ein!
Large task forces are to be set up, the industry adapts itself! (File ep-96-10-23.al, line 2723)
d) *jmd. etw. durch etw. einstellen – s.th. being established between s.o.*
Aus bestimmten Gründen stellten die beiden Fraktionen ihre Feindseligkeiten vorübergehend durch einen Waffenstillstand ein und vereinbarten untereinander vertrauensbildende Maßnahmen, doch der seit 1995 bestehende, brüchige Frieden führte in letzter Zeit zu erneuten Feindseligkeiten.
For some reason, a temporary cease-fire in the hostilities between the two fractions was established and certain confidence-building measures were agreed between them, but the fragile peace that existed through 1995 has recently given way to renewed hostilities. (File ep-96-09-18.al, line 1318)
e) *etw. darf nicht eingestellt werden – s.th. must not be ended*
Im Namen des Prinzips der Kontinuität, der Kohärenz, des Schutzes der finanziellen Interessen der Europäischen Union dürfen diese Beihilfen nicht eingestellt werden, die bereits gebunden sind. …
In the name of the principle of continuity, coherence and protecting the financial interests of the European Union, the aid that has already been committed must not be ended, … (File ep-00-06-15.al, line 2333)

Some translations, including appropriate constraints, should be obvious for the experienced lexicographer and therefore may be already included in the base lexicon. Examples are a) and b) which involve selectional constraints for the direct object: *eine PERSON einstellen – to hire a PERSON*, *eine MASCHINE einstellen – to adjust a MACHINE*.

Example c) is similar. A typical entry would be *SICH auf eine SITUATION einstellen – to adjust ONESELF to a SITUATION*. However, in contrast to this, the prepositional argument is eliminated in c). From this the question arises under what conditions this elimination takes place.

d) is an instance of so called *thematic divergence* (Dorr 1990), where one or several arguments change their thematic role during translation. (In this example the durch-PP takes the role of the direct object and the subject is realized as between-PP.) Example d) shows also a kind of complex *incorporation*, where a (verb-) role is eliminated as such during translation: Instead it appears as in-PP of the object and is modified by the former subject.

Example e) illustrates how through movement at the surface structure (in this case by extraposition of the relative clause) the correspondences between the partial phrases can be modified. It thereby motivates a representation that is independent of the surface form, abstracting away from specific positions. At the same time it raises a question regarding the extent of the relevant constraints: Is *einstellen* translated by *ending* in the context of *BEIHILFE* **darf** *nicht eingestellt werden*, or does the condition *BEIHILFE nicht eingestellt werden* suffice? Is the negation or the passive crucial for the choice of this translation? What exactly is the relevant selectional restriction for the surface subject, etc.?

It is not necessarily the case that the translation relations considered essential have a high occurrence frequency in the corpus. For example, relation b) appears less than 10 times in about 1200 occurrences of (inflected) forms of *einsetzen*, including forms with separated prefix, within 1.3 million Europarl sentences. However, the corpus often contains more specific translations, which as such have not yet been taken into account by the lexicographer. d) is a typical example.

The semi-automatic lexicon expansion aims at locating translation examples which are not subsumed by existing lexicon entries. It further aims at generalizing them in an appropriate way, to test the results against the analyses of the sentences containing instances of the lemma under consideration (here: *einstellen*), and to verify the respective translations.

3 Underspecified analysis

In principle an arbitrary formal grammar could be used as starting point to assign underspecified representations to (sets of) corresponding syntactic analyses of corpus sentences. Our approach uses *slot grammar* as implemented and used in the research version of the MT tool *translate*, because its dependency-style analyses are especially apt to be mapped into underspecified representations; besides this, appropriate procedures are already available or can easily be adapted to the specific needs of the corpus investigation that was described above [7] (cf. Eberle 2002).

For the representation of underspecified structures the system uses *flat underspecified discourse representation theory* (FUDRT; cf. Eberle 1997, 2004), which is an extension of UDRT (cf. Reyle 1993) that can deal with lexical and functional ambiguities.

In FUDRT lexemes are mapped into *labeled functional terms* that range over semantic representations and are evaluated (partly) depending on conditions connected to them. Evaluation may relate either to solving an ambiguity, or to assigning a more

[7] Regarding *translate* see `http://lingenio.de/English/Products/translation-software.htm`; for *slot grammar* see McCord 1989, 1991

fine-grained representation. A label identifies a subrepresentation of the complete sentence representation; it is decorated by the *distinguished discourse referent(s)* (DRFs) introduced by this subrepresentation.

During the process of compositionally constructing the sentence representation, complements and adjuncts are not applied to their arguments in the sense of a beta conversion in the lambda calculus; rather, they are recorded in the *functor set* of the base representation (of the noun, verb etc.). A set of constraints defines how far the possibilities to relate these partial representations with each other are restricted beyond the requirements inherent to the representations (i.e. given by their syntactic and logical type). This set of constraints may be empty.

Using 2.e as an example, (2.e$_{rep}$) illustrates the shape of FUDRS-representations. We skip all details of the representation which are irrelevant to the topic; in particular we skip representing the tense information. (L$_i$ are representations, and l$_i$ the pertaining labels):

(2.e$_{rep}$)

$$\begin{array}{l}
\qquad\qquad\qquad\left\{\begin{array}{l}
\mathrm{l}_{1_{e_1}}:\underline{\mathrm{in}}(\underline{\mathrm{name}}(\mathrm{w})\{\ \cdots\ \},\mathrm{L'}_{1_{e_1}}),\\[2ex]
\mathrm{l}_{2_s}:\underline{\mathrm{dürfen}}(\mathrm{L'}_{2_{e_2}}),\\[2ex]
\mathrm{l}_{3_{X_{@obj}}}:\underline{\mathrm{dies}}(\underline{\mathrm{beihilfe}}(\mathrm{x})\{\ \underline{\mathrm{bereits}}(\underline{\mathrm{gebunden}}(\mathrm{L'}_{3_{x'}}))\ \}),\\[4ex]
\mathrm{l}_{4_s}:\underline{\mathrm{nicht}}(\mathrm{L'}_{4_{e'}})
\end{array}\right\}\\
\mathrm{l}_{0_e}:\boxed{\begin{array}{l}\mathrm{e}\\ \mathrm{einstellen(e)}\\ \mathrm{theme(e)= X}\end{array}}\quad\{\}
\end{array}$$

(2.e$_{rep}$) consists essentially of three parts. These are, firstly, the *basic representation* which introduces an event of type *einstellen e*, and which is assigned the label l_0 which is decorated by the distinguished DRF, *e*; secondly, the *set of functors* that relate to this basic representation; thirdly, the *set of constraints* which is empty here. It may contain information about how the functors relate to each other or to the basic representation, beyond the implicit constraints from the syntax-semantics interface (linking information etc.) and the information about their logical type. Four functors are given in the representation: they sketch the semantic contributions of the *Im Namen*-PP, labeled l$_1$, of the modal embedding (*dürfen*), labeled l$_2$, of the direct object, labeled l$_3$, and the negation, labeled l$_4$. All these (sketches of) representations are decorated by appropriate DRFs and give indications about their internal structure. In the case of the direct object this structure takes the shape of a complete FUDRS with functor set etc., because of its rich modifying structure. In these representations, underlining marks an expression as functional in the sense described above. Next to lexical and scopal underspecification, FUDRS can represent *functional underspecification.* [8]

For example d) this means that the two possibilities of relating the contribution of the PP at the beginning of the sentence (*Aus bestimmten Gründen*) to the subsequent coordination, can be summarized in a single representation. Hereby the first interpretation relates the PP solely to *Einstellen der Feindseligkeiten*, whereas the second relates it to *Einstellen* and additionally to the second part of the coordination *Vereinbaren vertrauensbildender Maßnahmen.*

In our view such underspecified representations are essential to make the representation of sentences at the semantic level feasible: Fully specified representations would involve considerable disambiguation efforts, which makes them seem inappropriate for a corpus based approach. But there are also fundamental reasons why they should be avoided: Often it is not necessary to consider specific readings. And in cases where this is necessary, FUDRS allow partial disambiguations. They are also advantageous when searching for specific analyses as they make it possible to summarize several cases in a beneficial way.

4 Computation of transfer relations on representations

When FUDRS representations are constructed for source and target sentences, they must be related to each other in order to be able to extract the FUDRS contexts of the words or expressions that are of interest.

Hereby a decision must be made concerning the maximum allowed structural distance between conditions. [9] At the current stage we assume that for verbs it suffices to consider only the partial representation that contains the verbal representation as a basis. We further assume that informations from subordinate clauses, from adverbials, and from discourse relations (e.g. explanatory relations) also do not play a direct role and consequently can be omitted. [10] We illustrate these assumptions using example (2.d). Of relevance for the translation of *einstellen* is only the information from the first conjunct of the *und*-coordination. According to what was said previously it would be possible that the *aus*-PP appeared as a functor in a respective

[8] Further details of the representations can, for example, be found in Eberle (2004) but are not of importance for the current considerations.

[9] This should not be mixed up with information that has some more indirect influence on the translation process. For example, information that contributes to the specification of the text type, the general topic, or the subject area. The computation of such information has to be conducted in a different way and opens up a different strait of research, which is not under consideration here.

[10] For other parts of speech we assume similar constraints which are also based on experience.

FUDRS. However, as this PP contains a discourse relation (a justification, as can be derived from the semantic type of the internal argument) it is not taken into account. What remains is the following partial representation:

$$(2.\mathrm{d}_{repD})\qquad \left\{\begin{array}{l} \mathrm{l}_{1_X}{:}\underline{\mathrm{die}}(\underline{\mathrm{beide}}(\underline{\mathrm{fraktion}}(\mathrm{x}))),\\ \mathrm{l}_{2_Y}{:}\underline{\mathrm{ihre}}(\underline{\mathrm{feindseligkeit}}(\mathrm{y})),\\ \mathrm{l}_{3_{e'}}{:}\underline{\mathrm{vorübergehend}}(\mathrm{L'}_{3_{e'}}),\\ \mathrm{l}_{4''_e}{:}\underline{\mathrm{durch}}(\underline{\mathrm{waffenstillstand}}(\mathrm{z}),\mathrm{L'}_{4''_e}) \end{array}\right\}$$

$$\mathrm{l}0_e{:}\boxed{\begin{array}{l} \mathrm{e}\\ \mathrm{einstellen(e)}\\ \mathrm{agent(e)} = \mathrm{X}\\ \mathrm{theme(e)}= \mathrm{Y} \end{array}}$$

When assigning a partial representation of the representation of the English sentence as a translation to this, structural similarities and the knowledge about translation relations from the base dictionary are exploited. A candidate within the $\underline{\text{but}}(\underline{\text{and}}(\ldots\underline{\text{establish}}\ldots,\ \ldots\underline{\text{agree}}\ldots),\ldots\underline{\text{give way}}\ldots)$-structure is the first argument of the $\underline{\text{and}}(\_,\_)$-representation:

$$(2.\mathrm{d}_{repE})$$

$$\mathrm{l}_{0_e}{:}\boxed{\begin{array}{l} \mathrm{e}\\ \mathrm{establish(e)}\\ \mathrm{theme(e)}= \mathrm{z} \end{array}}\quad \left\{\begin{array}{l} \mathrm{l}_{7_z}{:}\underline{\mathrm{a}}(\underline{\mathrm{cease\text{-}fire}}(\mathrm{z})\ \ \left\{\begin{array}{l}\mathrm{temporary}\\ \hline \mathrm{l}_{1_e}{:}\underline{\mathrm{in}}(\underline{\mathrm{hostility}}(\mathrm{y}),\mathrm{L'}_{1_e}\ \ \{\,\mathrm{l}_{1_e}{:}\underline{\mathrm{between}}(\ldots\underline{\mathrm{faction}}(\mathrm{x}),\mathrm{L'}_{1_e})\,\}\end{array}\right\}\ \ )\\ \{\} \end{array}\right\}$$

Thereafter it is checked whether the candidate contains representations relating to translations of relevant words according to the base lexicon. Relevant words are typically those which introduce base representations.[11] Important words in (2.d_{repD}) are *Fraktion, Feindseligkeit, vorübergehend, Waffenstillstand.* Within the same level it is possible to refine appropriately according to the role hierarchies (subcategorized/non subcategorized; "obliqueness"-hierarchy etc.) This verification process is of course dependent on the set of translation relations in the base lexicon. However, it can be assumed that even in the case of a very limited coverage the correct correspondence should show a better agreement than other candidates. Furthermore, structural and surface knowledge gives additional clues.

It also depends on the relational knowledge provided by the base lexicon whether restructuring demands regarding the discovered representations and their correspondences can be compiled fully automatically or only in part. Assuming that the translations of the above mentioned "important" words are available, we obtain for the (2.d_{repD})-(2.d_{repE})-case in the notation used for FUDRS-transfer in the *translate-*

[11] Hereby the relative importance can be ranked in descending order w.r.t. the functor hierarchy.

system (cf. Eberle 2001) the following specification:

(2.d$_{repDE}$)

- l_0:$\underline{\text{einstellen}}$ $^{[\text{subj(n),obj(n)}]}$

C: d(vadv):l_1:$\underline{\text{vorübergehend}}$ & d(subj):l_2:$\underline{\text{fraktion}}$
& d(obj):l_3:$\underline{\text{feindseligkeit}}$ & d(prep($\underline{\text{durch}}$)):l_4:$\underline{\text{waffenstillstand}}$

τ: $\underline{\text{establish}}$ $^{[\emptyset,\text{obj(n)}:\tau(l_4)]}$
& τ(d-l_1)=τ(l_0)-d(obj)-d(nadj) & τ(d-l_3)=τ(l_0)-d(obj)-d(prep($\underline{\text{in}}$))
& τ(d-l_2)=τ(l_0)-d(obj)-d(prep($\underline{\text{in}}$))-d(prep($\underline{\text{between}}$))

(2.d$_{repDE}$) uses path designations to refer to partial structures and describes them using abbreviations of their base structures. Path equations are used to describe complex shifts. The position C (for *condition*) describes the preconditions for selecting the translation described under τ. This complex condition is the starting point for successive generalizations of the transfer statement, where generalizations are created by omission of paths as well as by semantic generalization of structures along the available hierarchy of semantic types.

The generalization which eliminates the adverbials and generalizes the constraints about the subject to the mere constraint of being of type AGENTIVE may, for example, be considered suitable, perhaps even the generalization which replaces the description of the direct object and of the durch-PP by the SITUATION-supertype (for events, states, etc.). For structural reasons it is not allowed to omit the durch-PP as it delivers the (obligatory) object of the translation.

Checking the suitability means to verify the translation prediction using the corresponding sentences of the corpus.

5 Summary and further steps

The proposed procedure for extending the dictionaries is based on two steps: The first step extends translation dictionaries using information from monolingual corpora and thus helps to solve the notorious problem of acquiring large enough and rich enough corpora for the extraction of translation relations. The second step uses rule-based analysis systems to automatize the construction of translation relations and their conditions for inspection by human users. A prerequisite for both steps is the availability of a small manually prepared lexical core which contains the most essential words of the base vocabulary.

Further steps include the empirical validation of the results of the overall system, which is currently being examined, and the further combination of rule-based analysis and statistical evaluation; most importantly, the integration of analytical knowledge into the statistical procedures used for the compilation of the base lexicon, and the installation of feedback loops that include automatic statistical tests with the aim of optimizing the degree of automatization of the analysis based dictionary expansion procedure that we have described.

References

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, John D. Lafferty, and Robert L.Mercer (1992). Analysis, Statistical Transfer, and Synthesis in Machine Translation. In *4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal.

Callison-Burch, Chris, Philipp Koehn, and Miles Osborne (2006). Improved Statistical Machine Translation Using Paraphrases. In *Proceedings NAACL-2006*.

Dorr, Bonnie (1990). Solving Thematic Divergences in Machine Translation. In *Proceedings of the 28th Annual Conference of the Association for Computational Linguistics*, 127–134, Pittsburgh, PA.

Eberle, Kurt (1997). Flat Underspecified Representation and its Meaning for a Fragment of German. Arbeitspapiere des Sonderforschungsbereichs 340 *Sprachtheoretische Grundlagen für die Computerlinguistik* 120, Universität Stuttgart, Stuttgart.

Eberle, Kurt (2001). FUDR-based MT, Head Switching and the Lexicon. In *Proceedings of the eighth Machine Translation Summit*, Santiago de Compostela.

Eberle, Kurt (2002). Tense and Aspect Information in a FUDR-Based German French Machine Translation System. In Hans Kamp and Uwe Reyle (eds.), *How we say WHEN it happens. Contributions to the theory of temporal reference in natural language*, 97–148, Tübingen: Niemeyer, ling. Arbeiten, Band 455.

Eberle, Kurt (2004). Flat Underspecified Representation and its Meaning for a Fragment of German. Habilitationsschrift, Universität Stuttgart, Stuttgart.

Emele, Martin C., Michael Dorna, Anke Lüdeling, Heike Zinsmeister, and Christian Rohrer (2000). Semantic-Based Transfer. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, 359–376, Berlin, Heidelberg, New York: Springer.

Gale, William A. and Kenneth W. Church (1993). A Program for Aligning Sentences in Bilingual Corpora. *Compuational Linguistics* 19:75–102.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Mass.: MIT Press.

McCord, Michael (1989). A New Version of Slot Grammar. Research Report RC 14506, IBM research division, Yorktown Heights.

McCord, Michael (1991). The Slot Grammar System. In Jürgen Wedekind and Christian Rohrer (eds.), *Unification in Grammar*, MIT-Press.

Och, Franz Josef and Hermann Ney (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the Annual Meeting of the ACL*, 295–302, Philadelphia, PA.

Och, Franz Josef and Hermann Ney (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29(1):19–51.

Rapp, Reinhard (1999). Automatic Identification of Word Translations from Unrelated English and German Corpora. In *Proceedings of the 37rd Meeting of the Association for Computational Linguistics*, 519–526, College Park, Maryland.

Rapp, Reinhard and Carlos Martin Vide (2007). Statistical Machine Translation without Parallel Corpora. In Georg Rehm, Andreas Witt, and Lothar Lemnitzer (eds.), *Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007*, Tübingen: Narr.

Reyle, Uwe (1993). Dealing with Ambiguities by Underspecification: Construction, Representation, and Deduction. *Journal of Semantics* 10(2):123–179.

Simard, Michel, G. Foster, and Pierre Isabelle (1992). Using Cognates to Align Sentences in Bilingual Corpora. In *Proceeedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'92)*, Montréal, Canada.

Tsuruoka, Yoshimasa and Jun'ichi Tsujii (2005). Chunk Parsing Revisited. In *Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT)*, 133–140, Vancouver.

Part III
Representation of lexical knowledge and text resources

The spanish version of WordNet 3.0

Ana Fernández-Montraveta, Gloria Vázquez and Christiane Fellbaum

Abstract. In this paper we present the Spanish version of WordNet 3.0. The English resource includes the glosses (definitions and examples) and the labelling of senses with WordNet identifiers. We have translated the synsets and the glosses to Spanish and alignment has been carried out at word level, whenever possible. The project has produced two interesting results: we have obtained a bilingual (Spanish and English) lexical resource for WordNet which will be available at no cost, as well as a parallel Spanish-English corpus annotated at word level with not only morphosyntactic information but also semantic information.

1 Introduction

In this paper[1] we present a lexical resource, the Spanish version of the English WordNet 3.0 (cf. Fellbaum 1998; Miller et al. 1990). This resource is composed of the translations of the English synsets into Spanish and a parallel annotated corpus with the definitions and the examples of each synset.

This corpus will be specially of interest since it will not only be a parallel corpus but will also be partially annotated in both languages with morphosyntactic and semantic information at word level. There are other English-Spanish corpora. In some of them, alignment is established at paragraph level (cf. Gelbukh et al. 2006) whereas in others it is at word level (CRATER Corpus (cf. McEnery et al. 1997), GRIAL Trilingual Corpus (cf. Castellón et al. 2005), ACREL Corpus (cf. Ramon 2004)). In all of these examples, as far as we know, annotation is limited to the morphosyntactic level if indeed there is any annotation at all.

We are certain that the information provided in this resource will be very useful for different automatic tasks within the domain of NLP, such as semantic annotation and disambiguation for Spanish or within the scope of applied linguistics to carry out contrastive studies in Spanish and English.

Corpus annotation is an arduous task and, in keeping with the precedent of other projects such as MultiSemCor (cf. Bentivogli et al. 2005; Ranieri et al. 2004), we based our strategy on reusing the work already carried out for the annotation of the English corpus. Thus, we have worked with the annotated glosses provided by the University of Princeton.[2] The English glosses were already annotated at both

[1] The present work was funded by project HUM2007-65267.

[2] The annotation of English glosses was carried out by Fellbaum's team and funded by DTO/IARPA.

morphological and semantic levels. From this annotated corpus, we translated the variants and the glosses into Spanish and changed the annotation when it was necessary because the morphosyntactic category did not correspond in both languages. Alignment has been carried out at word level, whenever possible, in order to keep the original annotation structure.

Below, we reproduce an example that shows the kind of information annotated in the glosses:

> deed: a notable achievement
>
> a notable [lemma = notable%1; pos = JJ ; SK= notable%3:00:00"] achievement [lemma = achievement%1; pos = 01; SK = achievement%1:04:00::)

As can be observed, the information annotated in the glosses includes the morphological tag (POS), the lemma and the WordNet sensekey of some of the words. From this structure, the resulting Spanish gloss is:

> hazaña: un logro notable
>
> un logro [lemma = logro%1; pos = 01; SK = logro%1:04:00::)] notable [lemma = notable%1; pos = JJ ; SK= notable%3:00:00"]

In the first stages of the project, we attempted to automate the translation process by using several different tools for this purpose. However, the quality of the translation was so poor and required so much manual editing that we discarded this possibility. Additionally, in order to keep the annotation structure and the alignment at word level we decided to translate as literally as possible whenever this kind of translation made sense.

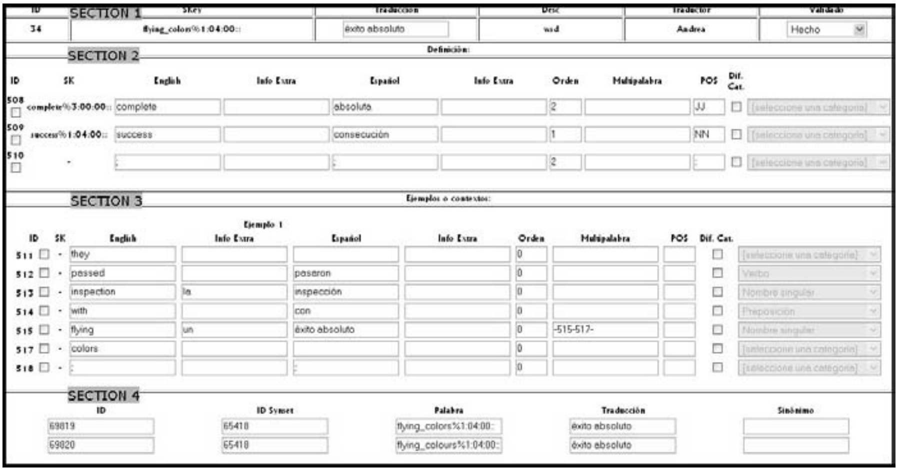
A team of 3 translators who are native Spanish speakers with a high command of English have been working part time on the translations. There has also been a coordinator, who is bilingual, in charge of resolving issues that have been considered problematic by the translators. The coordinator also has had to validate the translations done and make sure the structure and the annotation has been kept. All of this work has been done using an online interface.

Next we present further detail of the process of creation of the resource. First, we will show the interface in order to present a clearer picture of how alignment is established. Afterwards, we will see several examples illustrating the diverse cases found in the process of translation and parallelization.

The current resource is composed of 20,000 variants and 10,000 glosses (around 100,000 words). In the future, we expect to continue with the creation of this resource and finish with the translation of the 30,000 annotated glosses available in English. Our intention is to make the work totally available on the Internet. It represents an added value for the scientific community, since the only bilingual (Spanish and English) lexical resource for WordNet, the Spanish EuroWordNet, is only partially free.

2 The translation interface

We have created an interface to work on the translation of the variants, and, specially, of the glosses, the appearance of which is presented in the figure below:



SECTION 1

ID	SKey	Traducción	Desc	Traductor	Validado
34	flying_colors%1:04:00::	éxito absoluto	wsd	Andrea	Hecho

SECTION 2

Definición:

ID	SK	English	Info Extra	Español	Info Extra	Orden	Multipalabra	POS	Dif. Cat.	
508 ☐	complete%3:00:00::	complete		absoluta		2		JJ	☐	[seleccione una categoria]
509 ☐	success%1:04:00::	success		consecución		1		NN	☐	[seleccione una categoria]
510 ☐	-	:		:		2		:	☐	[seleccione una categoria]

SECTION 3

Ejemplos o contextos:

Ejemplo 1

ID		SK	English	Info Extra	Español	Info Extra	Orden	Multipalabra	POS	Dif. Cat.	
511	☐	-	they				0			☐	[seleccione una categoria]
512	☐	-	passed		pasaron		0			☐	Verbo
513	☐	-	inspection	la	inspección		0			☐	Nombre singular
514	☐	-	with		con		0			☐	Preposición
515	☐	-	flying	un	éxito absoluto		0	-515-517-		☐	Nombre singular
517	☐	-	colors				0			☐	[seleccione una categoria]
518	☐	-	:		:		0			☐	[seleccione una categoria]

SECTION 4

ID	ID Synset	Palabra	Traducción	Sinónimo
69819	65418	flying_colors%1:04:00:	éxito absoluto	
69820	65418	flying_colours%1:04:00:	éxito absoluto	

Figure 1. The translation and alignment interface.

The interface is divided into 4 sections. In the first section, identification information is displayed, such as the entry ID and the English synset variant (in the example, 'flying_colors%1:04:00::'). It also shows the field for the translation into Spanish of this variant ('traducción'), the name of the person who translates the gloss ('traductor') and the state of the translation ('validado'), which can be 'to be done', 'done', 'problematic' and 'validated'. First a translation is proposed ('done'); then it is revised and 'validated'. If translators are unsure how to translate a word or a part of the gloss, they label it as 'problematic', and if necessary a comment can be inserted.[3]

In sections 2 and 3, definition ('complete success') and examples ('they passed inspection with flying colors'), if any, are shown vertically. In both cases, the translation is carried out word by word in the column marked as Español in order to keep the annotations (morphosyntactic and semantic) of the English words whenever possible. Only the words in this field are parallel to the English version.

In the first column on the left ('ID'), the word (and punctuation mark) identifiers are shown. If a word is semantically annotated, this annotation is shown in the column beside (SK) by means of the WN sensekey assigned. In this example, all of the

[3] The field to insert comments is not visualized in the figure.

words of the English definition are labeled with this information (e.g. ‘complete’ carries the label ‘complete%3:00:00’ and ‘success’ the sensekey ‘success%1:04:00’).

The morphological annotation can be seen under the POS label. In the example seen, ‘complete’ is labeled as an adjective (JJ) and ‘success’ as a singular noun (01). Sometimes we need to change the category of a word when translating. To this aim we have created the box (Dif. Cat., different category). In this example this option has not been required but we will see an example later on.

One of the most common problems encountered when translating English to Spanish is that of word order. As can be seen in this case the order of the words that made up the definition is different in the two languages. The adjective always precedes the noun in English whereas in Spanish this is not always the case; in fact, it is usually the other way round. In order to account for order problems we have the column ‘Orden’. The value of this field is numeric and it expresses the order in which the translated words are to be shown in Spanish.[4]

The other three columns in this section correspond to ‘Info extra’ and ‘Multipalabra’ (extra information and multi-word respectively). As for the ‘multi-word’ field, it is used when it is not possible to make the correspondence between two concepts of the two languages word to word. If we take a look at the figure again, we will see that ‘flying colors’ has been established as a ‘multi-word’. In this case, the reason is that it is an expression in English, since its meaning is not compositional. Formally, this type of structure is created by linking the IDs of the words in English to just one field in the Spanish equivalent.

The field ‘extra information’ is used when more words are required in Spanish to express a meaning and will, therefore, not have a straight correspondence in the English annotated gloss. This could be true when, for example, we need a determiner in Spanish, as is the case of the image. As can be seen the articles ‘la’ and ‘un’ have been added to the Spanish structure and do not hold a link to any of the words in the English sentence.

Finally, in the fourth section of the interface we have four types of information: the first ID (further to the left) corresponds to the internal localization of the English WN database, the second ID is used to identify all the variants of a synset in the EWN; third, the two variants that are linked to the gloss and that are part of the same synset are visualized (in this gloss, in fact, there are two possible spellings of the same word), and, fourth, the translation to Spanish, that in this case is the same in both variants.

[4] If the number is 0, as in the example (section 3), it means that there are not any changes in the order of words in relation to English.

3 Different problems with parallelization

In this section we will briefly review some of the most common problems encountered when translating the text and aligning the Spanish and the English corpus. Obviously, we refer to mismatches in the translation that have a reflection on the alignment structure.

3.1 Adding functional words

One of the most common problems is the one just described: we need words in Spanish that do not have a direct counterpart in English. These elements would be left unlinked to any words in English but in a position between two elements of the sentence. Mostly, these are problems related to the different use of determiners in the two languages, as in the example above. As we can see in the examples below, this is also quite often the case with the use of the possessive adjectives (su) and with prepositions (de) that convey relations that in English are expressed by means of order.

(1) *Transporting alcoholic liquor for sale illegally.*
Transportando alcohólicas bebidas para venta ilegalmente.
'Transportar bebidas alcohólicas para su venta ilegal'.

(2) *Pocket-sized paperback book.*
Tamaño-de-bolsillo tapa-blanda libro
'Libro de tapa blanda de tamaño de bolsillo'.

3.2 Problems related to a different word order

As we have already pointed out this is an extremely common problem, and it usually affects the sequence 'adjective(s) plus noun', as in the example:

(3) *The experiencing of* emotional states.
La experimentación de emocionales estados.
'La experimentación de *estados emocionales*'.

At other times it is more complicated because what we essentially have are different structures in both languages.

3.3 Multi-word expressions

As we have said, the level at which alignment is established is the word since this is the level at which annotation in the English corpus is established. On occasion, this equivalence is not possible and links have to be established from an English expression to another expression in Spanish, as in the case of ‘flying colors’ and ‘éxito absoluto’, or to just one word, such as occurs in the following example:

(4) *There was* too much *for a single person to do.*
Había demasiado para una sola persona hacer
‘Había *demasiado* que hacer para una persona sola’.

Another common problem are clitic pronouns because they are graphically connected to the verb in Spanish when it is a gerund or an infinitive. In order to align them, we use the same mechanism we use with expressions, but a further morphological annotation process should analyze this form as a complex one formed by a verb and a pronoun.

It is also possible the contrary case, an English word is translated into more than one word in Spanish. This happens when English uses a synthetic process and Spanish an analytic one, for example, in the formation of some compounds (Eng. *trademark*, Spa. *marca registrada*) and also in some comparative and superlative forms:

(5) *The* biggest *overturn since David beat Goliath.*
El más-sorprendente resultado desde-que David ganó a-Goliat
‘El resultado *más sorprendente* desde que David ganó a Goliat’.

3.4 Different grammatical requirements

Given the fact that we are translating dictionary definitions and examples, the complexity of the grammatical structures to be translated is limited. Nevertheless there are some mismatches at this level that are worth remarking upon. A very common type is the use of gerund in English versus the use of infinitive in Spanish:[5]

(6) *A trap for* catching *rats.*
Una trampa para atrapando ratones
‘Trampa para *atrapar* ratones’.

Sometimes the differences between the two languages are even greater because there is not a complete correspondence at word level. In some instances, for example, a subordinate clause is required instead of an infinitive construction.

(7) *The Prohibition amendment made bootlegging profitable*
La Ley Seca hizo haciendo-contrabando rentable
‘La Ley Seca hizo *que el* contrabando *fuese* rentable’.

[5] (cf. Izquierdo 2006) for a analysis of the possibilities of translation of the -ing forms to Spanish.

We keep the alignment of every word that can be linked even though the grammatical structure is different. If necessary, the changes of category are codified, as in the case of the gerund *bootlegging*, which is expressed by a singular noun (*contrabando*) in Spanish. We used the field 'extra-information' to accommodate any extra words that do not have a counterpart in English. The words *que*, *el* and *fuese* are left unlinked to any English words but linked to the Spanish words that form their context.

Let's examine another example of grammatical mismatch between English and Spanish:

(8) *A miscalculation that recoils in its maker.*
Un error-de-cálculo que afecta a su realizador
'Un error de cálculo que afecta *al que lo* realiza'.

In this case, in Spanish a verb (*realize*) is used instead of a noun (*maker*) and, as a consequence, the resultant syntactic structure is quite different in both languages, since what in English is expressed by the possessive that determines the noun, in Spanish it is expressed by means of the subject (*al que*) and the object (*lo*) of the verb used.

3.5 Non-existence of a lexical counterpart

Sometimes, the synset we are translating belongs to a cultural reality (most of the time American) that does not have a straight counterpart in Spanish, at least lexically speaking, and thus the literal translation of the definition is impossible.

(9) *He came all the way around on William's* hit.
él llegó todo el camino alrededor en william'de golpe
'Llegó a la meta gracias al *golpe* de William.'

This example belongs to the domain of baseball. Baseball is barely known in Spain and thus the rules of the game are unknown to most people; so in the translation we decided to rephrase it to make it more understandable. 'Llegar a la meta' is more general than 'come all the way around' but the concept is the same; to reach a point that is the goal. So we have explained the meaning as much as possible keeping the pointers to the English semantic annotation ('hit, *golpe*'; SK: hit%1:04:03::).

Other examples of this type of mismatch are the well known verbs and deverbal nouns expressing manner in English. Manner in English can be expressed more generally than it is in Spanish where it usually requires a specification by means of an adjunct, as it can be seen in the case 'smack, *beso sonoro*'.

4 Conclusions

We have presented the results of the lexical and textual resource we have built aligning the WN glosses in English-Spanish. The lexical resource contains the translation of the English variants. Sometimes the equivalence is not a one-to-one since a language can have more synonyms for a concept than the other; thus, synsets have not necessarily the same number of variants in both languages. As for the glosses, they are annotated with POS and semantic information. They parallel WordNet 3.0 entries by keeping the annotation from this source whenever possible. We have tried to make translations as literal as possible, since, even though the morphosyntactic annotation is easily done, the semantic annotation is an arduous task and it is worthwhile to take advantage of work already completed.

Both resources will be very useful for NLP researchers working with Spanish since currently there is not any completely public resource for this language linked to any version of WordNet and, on the other hand, there are very few corpora for Spanish with annotation at semantic level. Also, it presents the added value of it being aligned to the English corpus and therefore it can contribute information in both languages from a contrastive perspective.

References

Bentivogli, Luisa, Emanuele Pianta, and Marcello Ranieri (2005). Multisemcor: an english-italian aligned corpus with a shared inventory of senses. In *Proceedings of the Meaning Workshop*, 90, Trento, Italy.

Castellón, Irene, Ana Fernández, and Gloria Vázquez (2005). Creación de un recurso textual para el aprendizaje del inglés. In *NOVATICA. Revista de la Asociación de técnicos de informática*, 51–54.

Fellbaum, Christiane (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Gelbukh, Alexander, Grigori Sidorov, and José Angel Vera-Félix (2006). Paragraph-level alignment of an english-spanish parallel corpus of fiction texts using bilingual dictionaries. In *Text, Speech and Dialogue*, 61–67, Berlin: Springer.

Izquierdo, Marlen (2006). Contrastive Analysis and Translation English-Spanish: functions of the English -ing form and its equivalents in Spanish. In *Multilingua*, http://multilingua.uib.no/marlen.page.

McEnery, Tony, Andrew Wilson, Fernando Sánchez-León, and Amalio Nieto-Serano (1997). Multilingual resources for European languages: Contributions of the Crater Project. In *Literary and Linguistic Computing*, 219–226.

Miller, George, Richard Beckwith, Christiane Fellbaum, David Gross, and Katherine Miller (1990). Introduction to WordNet: An On-line Lexical Database. In *International Journal of Lexicography*, 235–244.

Ramon, Noelia (2004). Building an English-Spanish Parallel Corpus for Teaching and Research: The ACTRES Project. In *Proceedings of the Sixth Teaching and Language Corpora Conference*.

Ranieri, Marcello, Emanuele Pianta, and Luisa Bentivogli (2004). Browsing Multilingual Information with the MultiSemCor Web Interface. In *Proceedings of the LREC 2004 Workshop The Amazing Utility of Parallel and Comparable Corpora*, 38–41, Lisbon, Portugal.

An OLIF-based open inflectional resource and yet another morphological system for German

Simon Clematide

Abstract. This paper describes the implementation of finite-state based, high precision morphological tools for the generation and analysis of open word classes based on the inflection classes for German of the Open Lexicon Interchange Format (OLIF). Productive compounding and derivations are treated by simple word formation rules. The latter is constrained by selective frequency checks over the web and corpora. Minimal lexicographic requirements (only stem and a numeric inflectional code) allow simple expandability and define a morphological abstraction layer which existing finite state morphological systems do not exhibit. Although a lot of lexical information is freely available for end users over the web, the same is not true for resources which will be used in NLP applications. Therefore, we initiate an open and shared morphological OLIF-based resource where we integrate material from sources which allows for such a term of use.

1 Introduction

The acquisition of morphological resources is commonly viewed as expensive in terms of “expert knowledge and labour” (Demberg 2007). Well, in fact it is expensive if it is done again and again by different academic researchers without sharing their resulting resources, and even more important, without a well-thought and well-agreed standard classification system which covers the needs of common text technology systems. For a highly inflected language as German, lemmatization and generation of inflected word forms is crucial for almost any text technological application. Simple and clear-cut interfaces for the coupling and extension of morphological and lexical resources are vital and should be based on standardized linguistic data categories. The EAGLES specification for German morpho-syntax (cf. EAGLES 1996) provides such a resource. For inflectional classes (in a very broad sense), the OLIF (Open Lexicon Interchange Format)[1] consortium has provided a list of “Recommended Values for OLIF Data Categories” for several languages including German (McCormick et al. 2004). In section 2, we described some more recent systems for German morphology. In section 3, we present our work in implementing a finite-state based morphological framework based on a minimal, but standard-oriented lexicographic interface.

[1] See `http://www.olif.net`.

2 Other works

Perera and Witte (2005) have built a self-learning system called `DurmLemmatizer` that induces a German full-form lexicon for nouns by processing raw text corpora[2]. Their linguistic processing is embedded in the GATE framework (cf. Cunningham et al. 2002) and restricted to a standard part-of-speech tagger (TreeTagger), a base NP chunker (JAPE), and their own case and grammatical number tagger based on Hidden Markov Model. Lemmatization is done by stripping off native German inflection suffixes, therefore plural forms involving umlaut as in “Ärzte (pl); Arzt (sg)” (*doctor*) can’t be treated correctly. In these and some other difficult cases, their algorithm inserts alternative possible lemma forms to gain recall (e.g. the possible lemma “*Öfen”,“*Öfe”, “*Öf” (*oven*)). These alternative forms may be reduced, if a further analysis appears with only one of the previously possible lemma forms. An assessment of the quality of the lemmatization based on this resource is more difficult than it may seem. Firstly, the evaluation results in their paper is based on a rather small lexicon with about 13’000 entries whereas the currently distributed resource contains about 84’000. Secondly, their own evaluation numbers need careful interpretation. They are gained against a subset of 88% of all noun occurrences where the TreeTagger was also able to produce a lemma. About 75% of the noun occurrences thereof are lemmatized by their system with a precision of around 95%. However, it’s unclear how they treat cases where the lexicon contains alternative lemmas – the current distribution of their lexicon has about 14’000 ambiguous lemmatizations.

The SOAP services from `http://wortschatz.uni-leipzig.de` allow the request for the generation of other word forms from a given one. This service is described as “For a given word form returns all other word forms of the same lemma”. The word form “geben” (*to give*) produces the output “gibt gab geben gegeben gebe gaben gäbe gibt’s Gibt gab’s” which makes obvious that only forms which are covered in the corpus are returned. The word form “lieben” (verb *to love* or adjective *dear*) seems to return adjective forms only: “lieber liebsten lieben liebe lieb liebste liebstes liebes liebster liebstem”. Although a verb and an adjective reading is returned by their base form service.

Geyken and Hanneforth (2006) present their German morphological analyzer based purely on finite state methods with weighted transitions[3]. The architecture of this system basically allows free combinations of the items from their stem (80’000 entries) and affix lexicon. About 1’000 morphotactic constraints (word grammar) restrict the possible combinations according to the language specific rules and limit morphological overanalyses. However, there are still lots of unwanted and irrelevant though possible morphological segmentations which one would like to get rid off.

[2] `http://www.ipd.uka.de/~durm/tm/lemma/`

[3] An online demo is available from `http://www.tagh.de`.

With the use of penalty weights associated with morphological boundaries and rare morphemes, an optimality ranking between competing analyses emerges from the analyses itself. Volk (1999) showed in the context of GERTWOL (cf. Koskeniemmi and Haapalainen 1996) that the heuristic "prefer simple analyses" is very effective in determining the intended lemma. Without weighted automata, one has to do this in a separate postprocessing filter.[4] Still, the weighted automata do not suppress unwanted analyses. The TAGH stem lexicon consists of complex entries because every stem alternation gives raise to a separate entry: E.G. the German verb "werfen" (*to throw*) needs the following lemma-stem pairs "werf:warf", "werf:werf", "werf:wirf", "werf:worf", "werf:würf" with their corresponding morphological features which determine the distribution of the stems in the inflectional paradigm. But there is also a lot of redundancy in this entries for the information which belongs to the lemma itself. The following two entries for past and past participle illustrate this point.

```
(werf:warf) [VIRREG VType=main PrefVerb=no Latinate=no StDef=yes St23SgInd=no
             StPret=yes StSubjI=no StSubjII=no StPartII=no StImpSg=no
             St23SgIndVowelChange=yes]
(werf:worf) [VIRREG VType=main PrefVerb=no Latinate=no StDef=yes St23SgInd=no
             StPret=no StSubjI=no StSubjII=no StPartII=yes StImpSg=no
             St23SgIndVowelChange=yes]
```

The TAGH system is optimized towards coverage.[5] For the 100 million word corpus "DWDS-Kerncorpus"[6] the authors give a coverage of 98.2%. Although no published quantitative evaluation on the correctness of the analyses is available, its effective use in two large scale and public lemmatization applications grants high quality.

Schmid et al. (2004) present a morphological analyser that recognizes derivation and composition. Stems may therefore be basic, derived or compounds. Affixes have the origin classes native, foreign, classical. They select their stems by word class features. An illustrating extract from the SMOR lexicon included in the SFST software distribution is shown below:

```
<Base_Stems>haus<PREF>:<><ge>ha:i<>:elt<V><base><nativ><VVPastStr>
<Base_Stems>haus<PREF>:<><ge>ha:ält<V><base><nativ><VVPres2t>
<Base_Stems>haus<PREF>:<><ge>halt<V><base><nativ><VVPP-en>
<Base_Stems>haus<PREF>:<><ge>halt<V><base><nativ><VVPres1>
<Base_Stems>g:bu:et:<><ADJ><base><nativ><AdjSup>
```

[4] Such a post-processing filter has an extreme low memory and processing cycle footprint if it's done using a standard UNIX flex tool as our own reimplementation of the original PERL code shows.

[5] However, on the demo web site they mention that rare word form (a threshold of 10 over a corpus of 500 million tokens) are omitted for efficiency reasons.

[6] `http://www.dwds.de`

```
<Base_Stems>g:bu:et:s<>:s<ADJ><base><nativ><AdjComp>
<Base_Stems>Roß:s<>:s<>:e<NN><base><nativ><NNeut/Pl>
```

The entries include structural (`<PREF>` “prefix”), morphotactic (`<nativ>`) and inflectional (`<VVPres2t>`) information. As in the case of TAGH, each stem alternation (e.g. `a:i`) is encoded by a separate lexicon entry. This is also true for suppletive gradation as “gut” (good), “besser” (better).

3 Architecture of mOLIFde

Other than the discussed SMOR or TAGH systems, our morphological system has minimal requirements for the lexicographic interface: An atomic stem[7] and an OLIF inflection code: E.G.

```
haus|halt 387
obig 531
Reichtum 111
```

For our internal lexical grammar, we strictly follow the EAGLES specification for German morpho-syntax (EAGLES 1996) which grants us compliance with STTS (Schiller et al. 1999) and documentation. We use the morpho-syntactic features and values verbatim (e.g. `"&pos" "=noun"`[8]) and serialize them top down according to the hierarchy presented in the standard. The raw EAGLES format and its corresponding shorter STTS representation look like

```
Reichtümern  &pos=noun&type=com&declin=no&numb=pl&case=dat&gend=masc&infl=--
Reichtümern  NN:Masc.Dat.Pl.*
```

3.1 The struggle with OLIF inflection categories

The recommended OLIF data categories for inflection codes contain more than 700 quite fine-graded word classes. For the open inflectional word classes, we find the following numbers: verbs (388), nouns (216), adjectives (34). These classes are more or less directly taken from the LOGOS machine translation system (cf. Scott 2004). To our knowledge, other lexical standardization initiatives (e.g. ISLE/MILE (Ide et al. 2003)) have not produced data category sets comparable to this list. Fig. 1

[7] The only exception is a boundary marker after separable verb prefixes that marks also the place for the insertion of “ge” in past participles.

[8] For a concise documentation on the syntax of the Xerox regular expression calculus see `http://www.xrce.xerox.com/competencies/content-analysis/fsCompiler/fssyntax.html`.

displays an extract of the noun inflection codes. Roughly said, they define a morphological abstraction layer which also covers some lexical and distributional informations needed for common text technological applications. Although the number of classes may be seen as high, coverage is not perfect.[9]

OLIF systematically shows separate classes for root verbs ("handeln" *to trade*), verbs with inseparable prefix ("behandeln" *to treat*), verbs with separable prefix ("herunterhandeln" *to beat down*), and verbs with a separable and an inseparable prefix ("wiederbehandeln" *to treat again*). The latter are quite uncommon as finite forms, however, adjectival use of past participles built out of them or nominalizations are more frequent. The German dictionary WAHRIG (Wahrig and Wahrig-Burfeind 2006) contains a list of 188 inflection paradigms for strong verbs, which would lead to an upper limit of 752 verb classes.

The high number of noun classes is mostly due to foreign words with foreign or alternate inflection paradigms ("Klima" *climate*, with 3 plural forms in nominative plural as "Klimata", "Klimate","Klimas") and the fact that every OLIF class has its determined gender even with identical inflection (e.g. "Vater" (*father*) masculine 51, "Kloster" (*convent*) neuter 141). There is also suppletive plural formation (e.g. the plural "Streuzuckersorten" for the uncountable German "Streuzucker" (*castor sugar*)) which may be practical for machine translation systems, but may seem idiosyncratic otherwise. Additional classes evolve from nouns with singular or plural forms only. Nouns with alternate paradigms get their own OLIF class which may lead to many additional classes when done consequently. Another more lexicographic question arises with nouns with alternate gender (often attributed to regional preferences, e.g. the masculine form "Gehalt" used in Austria in the sense of salary in contrast to the standard German neuter gender). And last but no least, spelling reforms of German have produced additional classes.

The linguistic characterisations of the different OLIF inflection classes are often sparse, as can be seen in Fig. 1. The use of arbitrary numbers as class identifiers may seem odd at first. The use of prototype lemmata in the style of "inflects like" should give a intuitive access to the classes. Still, an explicit explanation about the intended sense of a class would have made our work a lot easier. The example lemma itself may also be a source of confusion. For example, OLIF has an inflection class 105 `-s/-"e` exemplified by the lemma "Sonnenbrand" (*sunburn*) which therefore disallows "*Sonnenbrandes"[10]. Neither the Canoo language tools[11] nor WAHRIG support this limitation, and an exact Google search gives about 2'000 hits for "Son-

[9] Unfortunately, the integer IDs for the classes are not even unique across different part-of-speech. On the other hand, there are quite a few classes which are redundant, i.e. they cover the same phenomena.

[10] There exists a noun class 55 "Wunsch" (wish) `-es/-"e` that seems to enforce schwa in genitive singular.

[11] `http://www.canoo.net`

POS	Gender	Example	Inflects Like	Code
noun	feminine	Mutter	-/-" like Mutter/Mütter	53
noun	feminine	Hand	-/-"e like Hand/Hände	57
noun	feminine	Frau	-/-en like Frau/Frauen	64
adjective		arm	With umlaut and st in superlative like arm, ärmer, ärmst	96
verb		herausschinden	Irregular with separable prefix, like herausschinden - herausschund - herausgeschunden	645

Figure 1. Information contained in the OLIF inflection classes for German

nenbrands", but 8'000 for "Sonnenbrandes". There exist quite a few classes with overlapping or identical extension. The decision whether there is real redundancy has to be done painstakingly. In short, OLIF inflection codes were not as perfect as initially imagined. Along our development, we detected various problems and omissions which the OLIF consortium used to correct things according to our feedback.

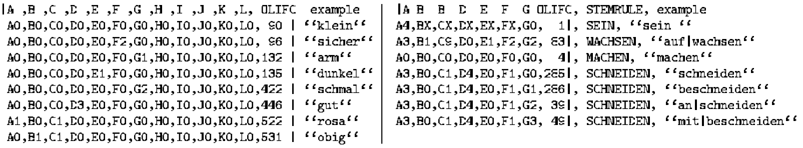
3.2 Our word-and-paradigm finite state morphology

Our system is implemented using the Xerox finite state tool `xfst` (cf. Beesley and Karttunen 2003). The benefits of transducers for morphology systems are common place now: Bidirectionality (generation and analysis), non-determinism (regular relations encode many-to-many mappings, i.e. a word form allows more than one analysis and the same morphological features may produce more than one word form), efficiency in processing time and memory.

One special feature of our system is the ability to generate word forms in a class based fashion. Our demo web service[12] generates any desired inflectional paradigm for a given lemma by specifying the corresponding OLIF inflection class. Though monolithic morphologic systems as SMOR or TAGH can generate, they are limited to their lexical content which can't be extended simply by a pair of stem and inflection class.

Finite state morphology engineering is either based on a two-level rule component as GERTWOL (Koskeniemmi and Haapalainen 1996), or on composition of replacements and restrictions since the invention of the replacement operator (Karttunen 1995). We decided to use the latter serial approach because our lemma lexicon does not contain stem alternation, and therefore a lot has to be done by rules to ensure the correct word forms.

[12] See `http://www.cl.uzh.ch/kitt/molif` for morphological generation and analyses.



```
|A ,B ,C ,D ,E ,F ,G ,H ,I ,J ,K ,L, OLIFC  example
 A0,B0,C0,D0,E0,F0,G0,H0,I0,J0,K0,L0, 90 | ‘‘klein‘‘
 A0,B0,C0,D0,E0,F2,G0,H0,I0,J0,K0,L0, 96 | ‘‘sicher‘‘
 A0,B0,C0,D0,E0,F0,G1,H0,I0,J0,K0,L0,132 | ‘‘arm‘‘
 A0,B0,C0,D0,E1,F0,G0,H0,I0,J0,K0,L0,135 | ‘‘dunkel‘‘
 A0,B0,C0,D0,E0,F0,G2,H0,I0,J0,K0,L0,422 | ‘‘schmal‘‘
 A0,B0,C0,D3,E0,F0,G0,H0,I0,J0,K0,L0,446 | ‘‘gut‘‘
 A1,B0,C1,D0,E0,F0,G0,H0,I0,J0,K0,L0,522 | ‘‘rosa‘‘
 A0,B1,C1,D0,E0,F0,G0,H0,I0,J0,K0,L0,531 | ‘‘obig‘‘
```

```
|A B  B  D  E  F  G OLIFC, STEMRULE, example
A4,BX,CX,DX,EX,FX,G0,  1|, SEIN, ‘‘sein ‘‘
A3,B1,C9,D0,E1,F2,G2, 83|, WACHSEN, ‘‘auf|wachsen‘‘
A0,B0,C0,D0,E0,F0,G0,  4|, MACHEN, ‘‘machen‘‘
A3,B0,C1,D4,E0,F1,G0,285|, SCHNEIDEN, ‘‘schneiden‘‘
A3,B0,C1,D4,E0,F1,G1,286|, SCHNEIDEN, ‘‘beschneiden‘‘
A3,B0,C1,D4,E0,F1,G2, 39|, SCHNEIDEN, ‘‘an|schneiden‘‘
A3,B0,C1,D4,E0,F1,G3, 49|, SCHNEIDEN, ‘‘mit|beschneiden‘‘
```

Figure 2. Extract of the matrix of linguistic features for adjectives and verbs

The huge number of inflection classes which had to be managed required a systematic specification approach with as much as possible automated reuse thereof. In the first place, every OLIF inflection class had to be reconstructed as a matrix of linguistic features.

Figure 2 shows some sample feature vectors. For adjectives, we have e.g. A0 = flectional, A1 = non-flectional; B0 = attributive and/or predicative use, B1 = attributive use only; C0 = unlimited gradation, C1=positive only; D3=irregular gradation stem; F2=optional elision of e in comparative forms; G0=no umlaut, G1=umlaut, G2=optional umlaut.

For verbs, we have e.g.: A=main verb class: A0=regular A4=special inflection A3=strong verb; B=special present forms: B0=no umlaut B1=umlaut; C=ablaut in past and past participle: C0=no change C1=ei-i-i C9=a-u-a; D: additional stem changes (consonant): D0=no change D4=d-tt; E=umlaut in past subjunctive: E1=normal umlaut; F=final sound classes: F1= dental (-d,-t) F2=sibilant (-s,-z) without -sch; G=verb prefix: G0=no prefix G1=inseparable prefix G2=separable prefix G3=both prefixes.

The inflection component for each adjective class has an architecture as depicted in Fig. 3. Similar architectures are used for verbs and nouns. In order to keep the manual writing of class-specific replacement rules consistent and short, two mappings are automatically built by processing the feature matrix.

- Feature macros (e.g. `AdjectiveMacroG2`) contain the union of every OLIF class tag exhibiting the corresponding feature. The restriction concerning attributive use can be written as:

```
define AdjectiveUseRestr [
    "&use"  => .#. [$. AdjectiveMacroB1] _ "=attr"
             ,  .#. [$. AdjectiveMacroB2] _ "=nattr" ];
```

- Class rules (e.g. `AdjectiveRule135`) contain the composition of general restrictions together with all the class specific feature rules (`AdjectiveRuleE1`) which have to be coded manually. The rule for feature E1 (deletion of “e” in attributive positive and comparative forms in lemmata as “dunkel” (dark)) looks like[13]:

```
define AdjectiveRuleE1 [
     [ $ [{el} "<DEGR/>"] ]  # precondition: ensure lemma is ending on -el
 .o. [ e -> 0 || _ l "<DEGR/>" ["<COMP/>"|"<POS/>"][$. ["&use" "=attr"]]]];
```

The precondition ensuring "-el" is only necessary for keeping generation of paradigms specific and discriminating, because it excludes any stem with feature E1 not ending on "-el" from producing word forms. This is essential if we try to induce the OLIF inflection class from full form lexica.

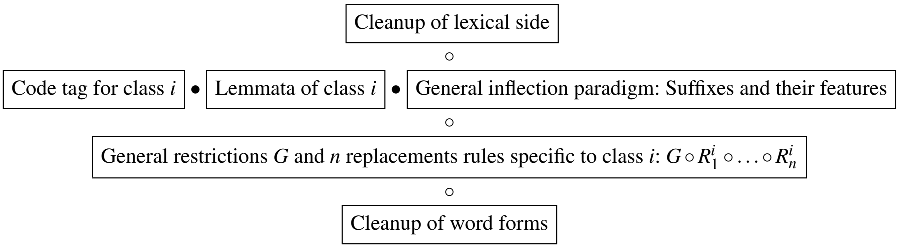

Cleanup of lexical side
∘
Code tag for class i
•
Lemmata of class i
•
General inflection paradigm: Suffixes and their features
∘
General restrictions G and n replacements rules specific to class i: $G \circ R^i_1 \circ \ldots \circ R^i_n$
∘
Cleanup of word forms

Figure 3. Main architecture of the mOLIFde inflection component for a single inflection class: ∘ means composition, • concatenation.

For class based generation, we need to keep the composed replacement rules separated from the lexicon. The composition of replacement rules (which are typically cyclic and reentrant) can quickly lead to huge transducers and long compilation times. A careful explicit definition of the lexical language and its composition to the rules has been proven critical to reach our goals. The compiler needs some hints where morphological values may appear and where they won't. An extreme example is the purely rule-based treatment for the stem "sein" (be) where we additionally to stem changes specify the real inflection paradigm.

```
...
.o. [{sei} "<VINFL/>" {en} -> {wär} "<VINFL/>" e        || _ ?* "&fin" "=fin"
     "&vf-m" "=konj" "&numb" "=sg" "&pers" ["=1"|"=3"] "&tense" "=past"]
.o. [{sei} "<VINFL/>" {en} -> {wär} "<VINFL/>" (e) {st} || _ ?* "&fin" "=fin"
    "&vf-m" "=konj" "&numb" "=sg" "&pers" "=2"  "&tense" "=past"]
.o. [{sei} "<VINFL/>" {en} -> {wär} "<VINFL/>" {en}     || _ ?* "&fin" "=fin"
    "&vf-m" "=konj" "&numb" "=pl" "&pers" ["=1"|"=3"] "&tense" "=past"]
.o. [{sei} "<VINFL/>" {en} -> {wär} "<VINFL/>" (e) t    || _ ?* "&fin" "=fin"
    "&vf-m" "=konj" "&numb" "=pl" "&pers" "=2" "&tense" "=past"]
...
```

[13] The gradation suffixes "er" and "st" are represented internally by abstract morphemes "<COMP/>" and "<POS/>" and realized in the cleanup step of the word form. This keeps the size of the composed transducers reasonable.

Without the lexical language the resulting transducer which generates all inflectional and non-inflectional forms, `xfst` gives the following properties: 110.1 Mb. 214706 states, 8886612 arcs, Circular. Composing the lexical language drastically reduces compilation time and size: 151.5 Kb. 812 states, 11507 arcs, Circular. Although this may still seem big, further composition with the lexicon entry "sein" results in a normal lexical transducer: 8.7 Kb. 280 states, 307 arcs, 34 paths.

3.3 Derivation, conversion, and compounding

Our lexicon doesn't provide origin information as SMOR. In contrast to compounding, derivation is a bounded process. Therefore, we can easily produce all derived lemmas[14] and validate them afterwards by frequency checks over web-based search engines and corpora[15]. Applying a threshold to the frequency counts gives us quite reliable results, although no systematic evaluation has yet been done. In the current state, we derive all verb forms with separable prefixes from a list of around 100 prefixes. For the frequency checks of this verbs, the past participle is a good choice. Productive and regular derivations which we would like to treat properly appear often in iterated suffixation (adjectives ending on "-ig" derive nouns on "-igkeit"). The corresponding OLIF inflection classes of the source stem and the derived stem can be predicted with high precision.

Productive noun compounding is done as in SMOR with inflected forms (nominative singular and plural, genitive singular) for the first element using the inflection suffix as the linking morpheme. This has to be enriched by feminine noun classes with linking elements "-s-" that are not part of their inflectional paradigm, as well as some nouns as "Schule" (*school*) where final "e" is deleted as in "Schulhaus" (*school building*).

The problem of overanalyses introduced by compounding is also present in our system. Within the finite state calculus we implemented optionally a method called "lexicon prioritizing" to effectively remove overanalyses in the lexical transducer which are already covered by the lexicon. First, we determine a transducer that has all word forms of analyses from simple lexicon entries which can be reanalyzed by compounds on one side, and on the other side the corresponding compound analyses we want to suppress. Second, we use the side with the compound analyses to remove them from the lexical side of the original transducer. The calculation for this operations takes some minutes for the current lexicon size (see Fig. 4) and it's the most expensive compilation step regarding memory consumption and processing time.

[14] Of course, conversion has also to be done. We have implemented a fix point computation that stops when conversion and derivation do not produce further new forms.

[15] The SOAP services from `http://wortschatz.uni-leipzig.de` are very useful for this.

3.4 An open OLIF-based German lexicon

The lack of open and shared high-quality morphological resources adapted for the use in text technological applications is a dissatisfying situation for a language as German. Although, there is currently an interest in the automatic learning of morphological segmentation Demberg (2007)[16], the results in the `DurmLemmatizer` lexicon show the difficulties of purely data-oriented boot-strapping approaches.

When we decided to adhere to the OLIF inflection classes, we had the aim to find preclassified entries which could be easily integrated. One hope was the lexicon of the OpenLogos[17] translation system which contains a huge relational database and which was the original source of the OLIF inflection classes. Unfortunately, we had some problems to access it and to take it apart. Currently we are in the process of integrating and mass validating its 165'000 lemmas into our resources we converted in the meantime. The number of lemmas is high, because conversion results as nominalized infinitives and deverbal adjectives are separately listed.

In the first time, we used the full form lexicon which can be exported from the public, but closed source Windows-based system Morphy (Lezius 2000) to induce the inflection classes. Our morphology produced the possible paradigms for each stem, then we compared the results with Morphy's paradigm, and tried to identify a single class. In the course of this work, we found several omissions and errors on our side as well as some peculiarities how Morphy treats the rare past subjunctive forms of strong verbs. For about 21'000 noun lemmas, 5'500 adjective lemmas, 4'000 verbs lemmas (without separable verb prefixes) a single class was identified. One interesting point of this resource in terms of analyses coverage is the tendency of Morphy to postulate a lot of singulare tanta nouns and non-gradable adjectives – although in many cases, it's morphologically sound to produce plural or comparative forms. The restrictions stem from semanto-lexicographic determinations of the words which normally takes place when word forms are coupled with specific meanings. The same kind of frequency checks we use for the validation of derived word forms, can be used to check and quantify the tendency for restricted use of such words.

Third, we used open bilingual resources[18] and extracted adjectives and nouns with frequent and regular suffixes. Classification validation can be done more quickly this way.

[16] `http://www.cis.hut.fi/morphochallenge2008`

[17] `http://logos-os.dfki.de`

[18] `http://www.dict.cc`

Category	Lexicon	Conversion	Derived	All
Verb	4745	0	17393	22138
Noun	20474	21987	15526	57987
Adjective	12173	43865	2997	59035
All:	37392	65852	35916	139160

Figure 4. The current distribution of lexical entries with derivative forms filtered by a threshold of 5 occurrences.

4 Conclusion

We think that a shared, simply extendable, and standard-based morphological resource for German fills a gap for text technology and lexicography. High precision lemmatization and generation of word forms should be standard techniques, self-learning systems may help to extend or optimize further. Huge and well supported corpora with application interfaces are an invaluable service therefore. The use of closed-source software for our morphological tools may seem inconsistent. However, our approach needed powerful and developer-friendly finite state tools already two years ago when the development started. For the finite part of the lexicon we have created an textual export into the open-source SFST tools. A current project will use our morphology in a web-service for generation of inflected forms for the automatic recognition of glossary entries in the OLAT[19] learning management system. [20]

References

Beesley, Kenneth R. and Lauri Karttunen (2003). *Finite-State Morphology: Xerox Tools and Techniques*. CSLI Publications.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan (2002). A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 168–175, University of Pennsylvania, URL `http://www.aclweb.org/anthology/P02-1022.pdf`.

Demberg, Vera (2007). A Language-Independent Unsupervised Model for Morphological Segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 920–927, Prague, Czech Republic: Association for Computational Linguistics, URL `http://www.aclweb.org/anthology/P/P07/P07-1116`.

EAGLES (1996). ELM-DE: EAGLES Specifications for German morphosyntax: Lexicon Specification and Classification Guidelines. electronic, URL `http://www.ilc.cnr.it/EAGLES96/pub/eagles/lexicons/elm_de.ps.gz`.

[19] `http://www.olat.org`

[20] Thanks to Thomas Kappeler and Luzius Thöny for implementing the verb and noun part of the system.

Geyken, Alexander and Thomas Hanneforth (2006). *Finite-State Methods and Natural Language Processing, 5th International Workshop, FSMNLP 2005, Helsinki, Finland, September 1-2, 2005. Revised Papers*, chapter TAGH: A Complete Morphology for German Based on Weighted Finite State Automata, 55–66. Springer, URL http://dx.doi.org/10.1007/11780885_7.

Ide, Nancy, Alessandro Lenci, and Nicoletta Calzolari (2003). RDF Instantiation of ISLE/MILE Lexical Entries. In *Proceedings of the ACL 2003 workshop on Linguistic annotation*, 30–37, Morristown, NJ, USA: Association for Computational Linguistics, doi:http://dx.doi.org/10.3115/1119296.1119301.

Karttunen, Lauri (1995). The Replace Operator. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 16–23, Cambridge, Mass, URL http://www.aclweb.org/anthology/P95-1003.pdf.

Koskeniemmi, Kimmo and Mariikka Haapalainen (1996). GERTWOL – Lingsoft Oy. In Roland Hausser (ed.), *Linguistische Verifikation : Dokumentation zur Ersten Morpholympics 1994*, number Band 34 in Sprache und Information, 121–140, Tübingen: Niemeyer.

Lezius, Wolfgang (2000). Morphy - German Morphology, Part-of-Speech Tagging and Applications. In Ulrich Heid, Stefan Evert, Egbert Lehmann, and Christian Rohrer (eds.), *Proceedings of the 9th EURALEX International Congress*, 619–623, Stuttgart.

McCormick, Susan M, Christian Lieske, and Alexander Culum (2004). OLIF v.2: A Flexible Language Data Standard. URL http://www.olif.net/documents/OLIF_Term_Journal.pdf.

Perera, Praharshana and Rene Witte (2005). A Self-Learning Context-Aware Lemmatizer for German. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 636–643.

Schiller, Anne, Simone Teufel, and Christine Stöckert (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). URL http://www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-1999.pdf.

Schmid, Helmut, Arne Fitschen, and Ulrich Heid (2004). SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection. In *Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004)*, 1263–1266.

Scott, Bernard (Bud) (2004). The Logos Model: An Historical Perspective. *Machine Translation* 18:1–72, URL http://dx.doi.org/10.1023/B:COAT.0000021745.20402.59.

Volk, Martin (1999). Choosing the Right Lemma when Analysing German Nouns. In *Multilinguale Corpora: Codierung, Strukturierung, Analyse. 11. Jahrestagung der GLDV*, 304–310, Frankfurt.

Wahrig, Gerhard and Renate Wahrig-Burfeind (eds.) (2006). *Wahrig Deutsches Wörterbuch: mit einem Lexikon der Sprachlehre*. Gütersloh: Wissen Media Verlag, 8. edition.

Tools for exploring GermaNet in the context of cl-teaching

Irene Cramer and Marc Finthammer

Abstract. Word nets, such as Princeton WordNet or GermaNet, are resources organizing a (more or less extensive) fraction of the vocabulary of a language according to lexical semantic relations. Such resources are widely used in natural language processing (NLP) and computational linguistics (CL) both for research and teaching purposes. While several graphical user interfaces (GUI) exist for Princeton WordNet – some of which are also available online – GermaNet still lacks such utilities. In this paper we describe two GUI-based tools meant to facilitate the navigation through and exploration of GermaNet. Both are freely available for download from our project web page (`www.hytex.info`). We additionally discuss ways of deploying these tools in teaching. We argue that the GUI-based access to word nets, which can be regarded as a fundamental resource in CL and NLP, enhances the students' understanding of basic lexical semantic concepts, computational semantics and lexicography.

1 Motivation

Word nets are lexical semantic resources modeled according to the principles introduced in Princeton WordNet (e.g. Fellbaum 1998). The central idea of word nets is to group synonymous lexical units, also including compounds and multi-word-units, into so-called synsets (synonym sets) and link them according to lexical semantic relations, such as hyponymy, meronymy, antonymy etc. Currently, Princeton WordNet (Version 3.0) contains approximately 150,000 synsets[1] and approximately 200,000 lexical units. The conceptual design and the resource itself are upgraded continuously – e.g. over the past years proper names have been added and tagged accordingly (Miller and Hristea 2006) and non-classical, i.e. psycho-linguistically motivated, link types have been included as an additional layer of relations (Boyd-Graber et al. 2006). Many NLP-applications, such as information retrieval and information extraction (e.g. Mandala et al. 1998) or word sense disambiguation (e.g. Banerjee and Pedersen 2002), highly rely on word nets as a (lexical) semantic resource [2]. Therefore, in recent years, word nets have been developed for many languages, e.g. in the context of EuroWordNet (Vossen 1998) for seven European languages, and

[1] Please refer to `http://wordnet.princeton.edu/man/wnstats.7WN` for more information.

[2] Cp. Fellbaum (1998), Kunze (2001), and the Proceedings of the Global WordNet Conferences e.g. Tanács et al. (2008)

connected via the so-called ILI[3]. GermaNet, the German counterpart of Princeton WordNet, which has been developed since 1997 at the University of Tübingen, currently (Version 5.1) consists of 58,000 synsets and 82,000 lexical units[4].

As word nets constitute a fundamental resource in many NLP-applications, they should also play a major role in CL curricula and be carefully introduced in courses on e.g. computational semantics and NLP resources. In addition to the modeling and structure of word nets, students should be familiarized with algorithms for the calculation of semantic relatedness, similarity, and distance (cp. Budanitsky and Hirst 2006; Patwardhan and Pedersen 2006) – both from a theoretical and a practical point of view. Such algorithms are regarded as a fundamental component in various NLP-applications, such as text summarization (e.g. Barzilay and Elhadad 1997), malapropism recognition (e.g. Hirst and St-Onge 1998), automatic hyperlink generation (e.g. Green 1999), question answering (e.g. Novischi and Moldovan 2006), and topic detection/topic tracking (e.g. Carthy 2004). And even for traditional courses on e.g. semantics, word nets offer interesting options. Typically, semantic relations are introduced providing a few more or less plausible examples (e.g. `Rappe`, Engl. black horse, is a hyponym of horse). In contrast, Princeton WordNet[5] and GermaNet offer plenty of illustrative material, since they both cover a wide range of lexical units connected via semantic relations. While Princeton WordNet already exhibits several GUI-based interfaces, some of which are also available online [6], GermaNet still lacks such utilities. This might have two causes: firstly, the research community working with German data is much smaller than the one working with English; secondly, some word nets are subject to particular license restrictions[7]. In addition, GermaNet differs form Princeton WordNet with respect to some modeling aspects; therefore, tools implemented for WordNet cannot be adopted for GermaNet in its current state. While implementing a lexical chainer – called GLexi, (cf. Cramer and Finthammer 2008) – for German specialized domain corpora, Finthammer and Cramer (2008) implemented two GUI-based tools for the exploration of GermaNet.

Sections 2 and 3 introduce these tools and their basic features. Most researchers working with GermaNet share the same experience of getting lost in the rich structure of its XML-representation. Thus, the GUI-based tools implemented by Finthammer and Cramer (2008) are meant to help both researchers and students explore

[3] ILI stands for interlingual index. Please refer to `http://www.illc.uva.nl/EuroWordNet/` for more information.

[4] Please refer to `http://www.sfs.uni-tuebingen.de/lsd/` for more information.

[5] Princeton WordNet also features glosses explaining the meaning of a lexical unit and example sentences; it thus represents a full-fledged digital dictionary, which could be used in various application scenarios, e.g. as an interesting and innovative resource in classes of (computational) lexicography.

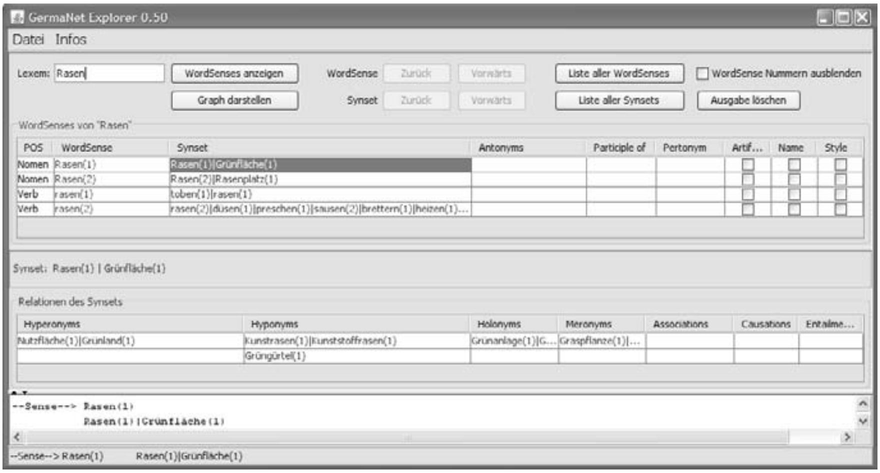
[6] E.g. WordNet Browser (`http://wordnet.princeton.edu/perl/webwn`) or Vocabulary Helper (`http://poets.notredame.ac.jp/cgi-bin/wn`).

[7] Please refer to `http://www.sfs.uni-tuebingen.de/lsd/` for more information on this issue.

GermaNet[8]. In this paper, we also discuss possibilities of how to utilize the tools in CL courses, especially practical sessions on lexical/computational semantics or computational lexicography. We have already used the tools in an annotation experiment (Cramer et al. accepted) with first-year and second-year students. We found that students employing the two GUI-based tools need less training than the students employing the XML-representation of GermaNet only. We also think that the GUI-based GermaNet interfaces might enhance the students' understanding of basic lexical semantic concepts. We therefore sketch some ideas of practical sessions introducing GermaNet and semantic relatedness measures drawing on the two tools in the following sections.

2 GermaNet Explorer

GermaNet Explorer, of which a screenshot is shown in Figure 1, is a tool for exploration and retrieval. Its most important features are: the word sense retrieval function (Figure 2) and the structured presentation of all semantic relations pointing to/from the synset containing the currently selected word sense (Figure 3). The GermaNet Explorer also provides a visual, graph-based navigation function: a synset (in Figure 4 [Rasen, Grünfläche] Engl. lawn) is displayed in the center of a navigation



Lexem: Rasen

□ WordSense Nummern ausblenden

WordSenses von "Rasen"

POS	WordSense	Synset	Antonyms	Participle of	Pertonym	Artif...	Name	Style
Nomen	Rasen(1)	Rasen(1)|Grünfläche(1)				□	□	□
Nomen	Rasen(2)	Rasen(2)|Rasenplatz(1)				□	□	□
Verb	rasen(1)	toben(1)|rasen(1)				□	□	□
Verb	rasen(2)	rasen(2)|düsen(1)|preschen(1)|sausen(2)|brettern(1)|heizen(1)...				□	□	□

Synset: Rasen(1) | Grünfläche(1)

Relationen des Synsets

Hyperonyms	Hyponyms	Holonyms	Meronyms	Associations	Causations	Entailme...
Nutzfläche(1)|Grünland(1)	Kunstrasen(1)|Kunststoffrasen(1)	Grünanlage(1)|G...	Graspflanze(1)|...			
	Grüngürtel(1)					

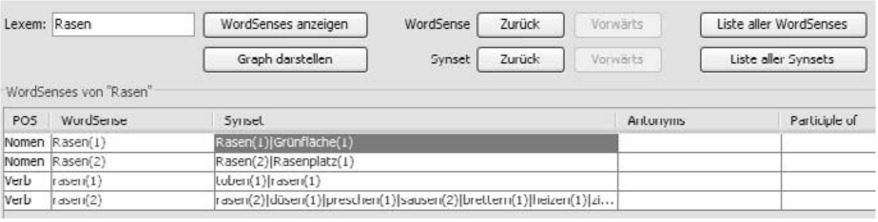
```
--Sense--> Rasen(1)
           Rasen(1)|Grünfläche(1)
```

--Sense--> Rasen(1)   Rasen(1)|Grünfläche(1)

Figure 1. Screenshot GermaNet Explorer

[8] The tools have been implemented in the context of the DFG-funded project HyTex and are freely available for download from our project web page (`www.hytex.info`).

graph surrounded by its direct semantically related synsets, such as hypernyms (in Figure 4 [`Nutzfläche, Grünland`]) above the current synset, hyponyms (in Figure 4 [`Kunstrasen, Kunststoffrasen`] and [`Grüngürtel`]) below, holonyms (in Figure 4 [`Grünanlage, Gartenanlage, Eremitage`]) to the left, and meronyms (in Figure 4 [`Graspflanze, Gras`]) to the right. In order to navigate the graph representation of GermaNet, one simply clicks on a related synset, in other words one of the rectangles surrounding the current synset shown in Figure 4. Subsequently, the visualization is refreshed: the selected synset moves into the center of the displayed graph, and the semantically related synsets are updated accordingly. In addition, the GermaNet Explorer features a representation of all synsets, which is illustrated in Figure 5. It also provides retrieval, filter, and sort functions (Figure 6). Moreover, the GermaNet Explorer exhibits the same functions as shown in Figures 5 and 6 with a similar GUI for the list of all word senses. We found that these functions, both for the word senses and the synsets, provide a very detailed insight into the modeling and structure of GermaNet. E.g. in a hands-on session of a (computational) semantics course, using the GermaNet Explorer students can (visually) examine lexical semantic relations for a (relatively) large fraction of the German vocabulary. While exploring sub-sets of lexical units, they can also compare their own intuition as well as the intuition of a group of German native speakers (namely, their fellow students) about the semantic relations between these lexical units with the modeling present in GermaNet. Potentially observed differences between their own intuition, the intuition of their fellow students, and GermaNet will certainly raise their awareness

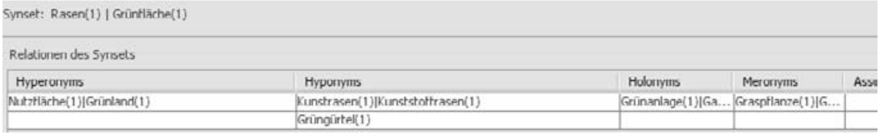


Lexem: Rasen

WordSenses von "Rasen"

POS	WordSense	Synset	Antonyms	Participle of
Nomen	Rasen(1)	Rasen(1)|Grünfläche(1)		
Nomen	Rasen(2)	Rasen(2)|Rasenplatz(1)		
Verb	rasen(1)	toben(1)|rasen(1)		
Verb	rasen(2)	rasen(2)|düsen(1)|preschen(1)|sausen(2)|brettern(1)|heizen(1)|zi...		

Figure 2. Screenshot GermaNet Explorer: Retrieval Functions



Synset: Rasen(1) | Grünfläche(1)

Relationen des Synsets

Hyperonyms	Hyponyms	Holonyms	Meronyms	Ass
Nutzfläche(1)|Grünland(1)	Kunstrasen(1)|Kunststoffrasen(1)	Grünanlage(1)|Ga...	Graspflanze(1)|G...	
	Grüngürtel(1)			

Figure 3. Screenshot GermaNet Explorer: Relations Pointing to/from Current Synset

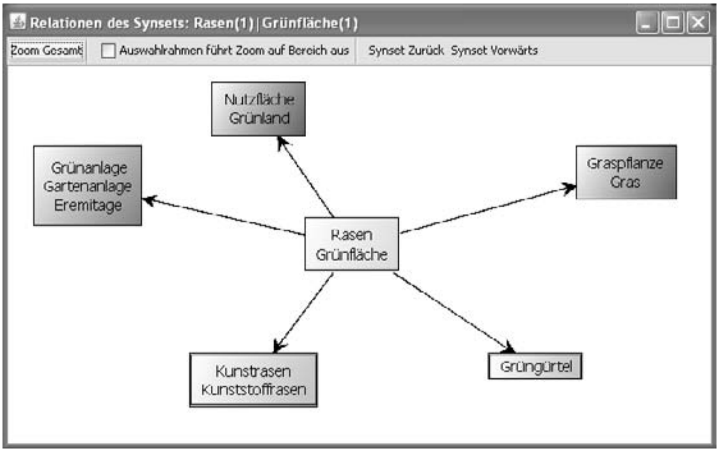
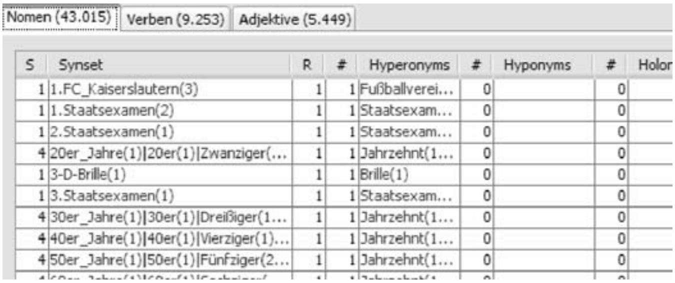


Figure 4. Screenshot GermaNet Explorer: Visual Graph Representation

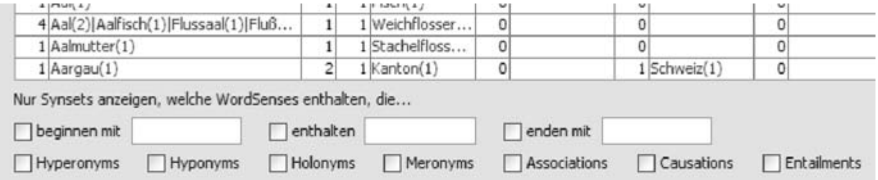


S	Synset	R	#	Hyperonyms	#	Hyponyms	#	Holor
1	1.FC_Kaiserslautern(3)	1	1	Fußballverei...	0		0	
1	1.Staatsexamen(2)	1	1	Staatsexam...	0		0	
1	2.Staatsexamen(1)	1	1	Staatsexam...	0		0	
4	20er_Jahre(1)|20er(1)|Zwanziger(...	1	1	Jahrzehnt(1...	0		0	
1	3-D-Brille(1)	1	1	Brille(1)	0		0	
1	3.Staatsexamen(1)	1	1	Staatsexam...	0		0	
4	30er_Jahre(1)|30er(1)|Dreißiger(1...	1	1	Jahrzehnt(1...	0		0	
4	40er_Jahre(1)|40er(1)|Vierziger(1)...	1	1	Jahrzehnt(1...	0		0	
4	50er_Jahre(1)|50er(1)|Fünfziger(2...	1	1	Jahrzehnt(1...	0		0	

Figure 5. Screenshot GermaNet Explorer: List of All GermaNet Synsets

of lexical semantic concepts and the challenge of building such a resource consistently. Last but not least, simple corpus-based methods to extract semantic relations (such as the well-known Hearst patterns, cf. Hearst 1992) may be compared with relations in GermaNet. An example is shown in Figure 7. Moreover, by contrasting relations of the same type, the students can learn to discern differences in relation strength and semantic distance (cf. Boyd-Graber et al. 2006)[9]. Examples are shown in Figures 8 and 9. Obviously, the modeling of the synsets containing terminology,

[9] Boyd-Graber et al. (2006) cite the following as an example: "It is intuitively clear that the semantic distance between the members of hierarchically related pairs is not always the same. Thus, the synset [run] is a subordinate of [move], and [jog] is a subordinate of [run]. But [run] and [jog] are semantically much closer than [run] and [move]."



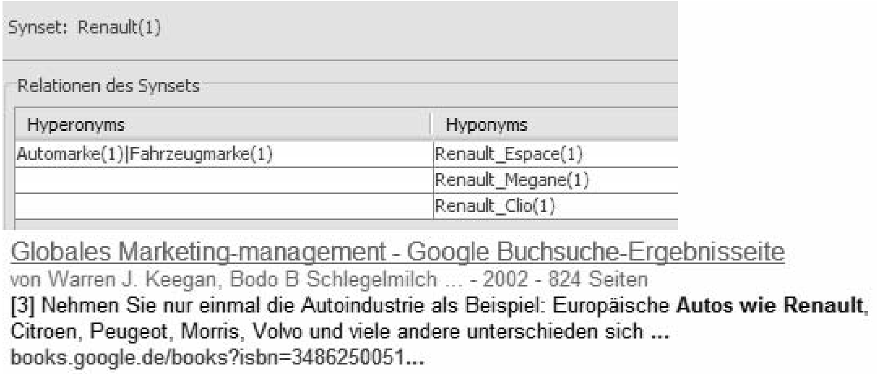
1	Aal(1)	1	1	Fisch(1)	0		0		0	
4	Aal(2)|Aalfisch(1)|Flussaal(1)|Fluß...	1	1	Weichflosser...	0		0		0	
1	Aalmutter(1)	1	1	Stachelfloss...	0		0		0	
1	Aargau(1)	2	1	Kanton(1)	0		1	Schweiz(1)	0	

Nur Synsets anzeigen, welche WordSenses enthalten, die...

□ beginnen mit □ enthalten □ enden mit

□ Hyperonyms □ Hyponyms □ Holonyms □ Meronyms □ Associations □ Causations □ Entailments

Figure 6. Screenshot GermaNet Explorer: List of All GermaNet Synsets: Filter and Search Functions



Synset: Renault(1)

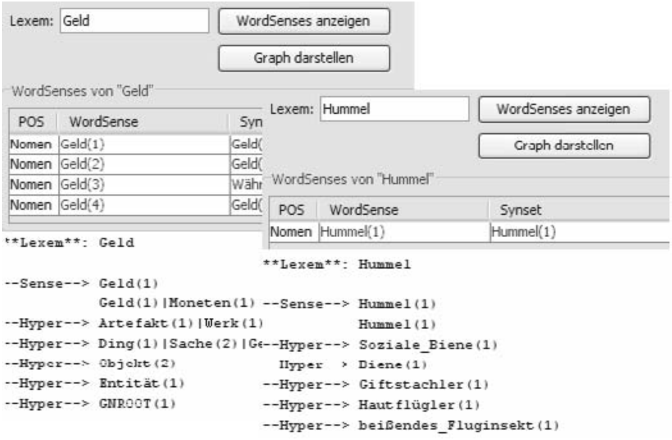
Relationen des Synsets

Hyperonyms	Hyponyms
Automarke(1)|Fahrzeugmarke(1)	Renault_Espace(1)
	Renault_Megane(1)
	Renault_Clio(1)

Globales Marketing-management - Google Buchsuche-Ergebnisseite
von Warren J. Keegan, Bodo B Schlegelmilch ... - 2002 - 824 Seiten
[3] Nehmen Sie nur einmal die Autoindustrie als Beispiel: Europäische **Autos wie Renault**, Citroen, Peugeot, Morris, Volvo und viele andere unterschieden sich ...
books.google.de/books?isbn=3486250051...

Figure 7. Hyponym Relation: GermaNet vs. Pattern-Based Corpus Approach

such as *bumble-bee*, is much more fine-grained than the one of synsets containing general concepts, such as *money*. The synsets of `fliegen` (Engl. to fly) and `kauen` (Engl. to chew) are both directly connected with `schwingen/oszillieren` (Engl. to oscillate), which implies that it only takes two steps from *fly* to *chew*. We assume that these and similar examples can improve the students' understanding of the structure and potential shortcomings of word nets in general and GermaNet in particular. Finally, even the often criticized lack of glosses in GermaNet may be used productively in order to discuss (sometimes subtle) differences in the meaning of lexical units or synsets. I.e. the meaning of synsets and lexical units can be retraced on the basis of the lexical semantic relations modeled in GermaNet. As an experiment, the thus extracted information can again be exploited by the students to write glosses and example sentences.



Lexem: Geld

WordSenses von "Geld"

POS	WordSense	Syn
Nomen	Geld(1)	Geld(
Nomen	Geld(2)	Geld(
Nomen	Geld(3)	Währ
Nomen	Geld(4)	Geld(

```
**Lexem**: Geld

--Sense--> Geld(1)
           Geld(1)|Moneten(1)
--Hyper--> Artefakt(1)|Werk(1)
--Hyper--> Ding(1)|Sache(2)|Ge
--Hyper--> Objekt(2)
--Hyper--> Entität(1)
--Hyper--> GNROOT(1)
```

Lexem: Hummel

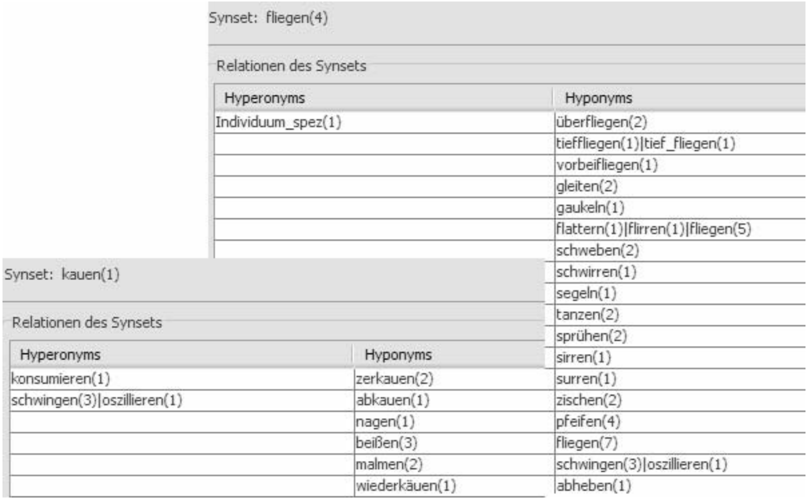
WordSenses von "Hummel"

POS	WordSense	Synset
Nomen	Hummel(1)	Hummel(1)

```
**Lexem**: Hummel

--Sense--> Hummel(1)
           Hummel(1)
--Hyper--> Soziale_Biene(1)
--Hyper--> Biene(1)
--Hyper--> Giftstachler(1)
--Hyper--> Hautflügler(1)
--Hyper--> beißendes_Fluginsekt(1)
```

Figure 8. Coarse-Grained vs. Fine-Grained Modeling of Synsets



Synset: fliegen(4)

Relationen des Synsets

Hyperonyms	Hyponyms
Individuum_spez(1)	überfliegen(2)
	tiefffliegen(1)|tief_fliegen(1)
	vorbeifliegen(1)
	gleiten(2)
	gaukeln(1)
	flattern(1)|flirren(1)|fliegen(5)
	schweben(2)
	schwirren(1)
	segeln(1)
	tanzen(2)
	sprühen(2)
	sirren(1)
	surren(1)
	zischen(2)
	pfeifen(4)
	fliegen(7)
	schwingen(3)|oszillieren(1)
	abheben(1)

Synset: kauen(1)

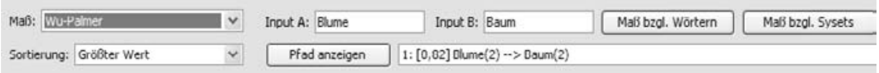
Relationen des Synsets

Hyperonyms	Hyponyms
konsumieren(1)	zerkauen(2)
schwingen(3)|oszillieren(1)	abkauen(1)
	nagen(1)
	beißen(3)
	malmen(2)
	wiederkäuen(1)

Figure 9. Two Steps form *Fly* to *Chew* Via *Oscillate*

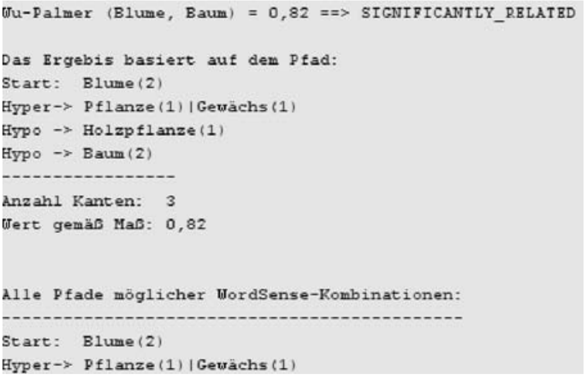
3 GermaNet Pathfinder

As mentioned in Section 1, the calculation of semantic relatedness, similarity, and distance plays a crucial role in many NLP-applications. Those measures express how much two words have to do with each other; they are extensively discussed in the literature (e.g. Budanitsky and Hirst 2006). Many measures have already been investigated and implemented for Princeton WordNet (e.g. Patwardhan and Pedersen 2006), however, there are only a few publications addressing measures based on GermaNet (e.g. Finthammer and Cramer 2008; Gurevych and Niederlich 2005). The GermaNet Pathfinder constitutes a GUI-based tool which has been developed as a central component of the lexical chainer GLexi (Cramer and Finthammer 2008). It implements eleven semantic measures – eight GermaNet-based[10] and three Google-based[11] ones – and integrates all measures into a common Java-API. The GermaNet Pathfinder additionally features a GUI meant to facilitate the intellectual analysis of semantic distance between given pairs of synsets or lexical units with respect to one semantic measure, a subset, or all. In short, the GermaNet Pathfinder exhibits the



Maß: Wu-Palmer
Input A: Blume
Input B: Baum
Sortierung: Größter Wert
1: [0,82] Blume(2) --> Baum(2)

Figure 10. Screenshot GermaNet Pathfinder



```
Wu-Palmer (Blume, Baum) = 0,82 ==> SIGNIFICANTLY_RELATED

Das Ergebis basiert auf dem Pfad:
Start:  Blume(2)
Hyper-> Pflanze(1)|Gewächs(1)
Hypo -> Holzpflanze(1)
Hypo -> Baum(2)
-----------------
Anzahl Kanten:  3
Wert gemäß Maß: 0,82


Alle Pfade möglicher WordSense-Kombinationen:
--------------------------------------------
Start:  Blume(2)
Hyper-> Pflanze(1)|Gewächs(1)
```

Figure 11. Screenshot GermaNet Pathfinder: Shortest Path and All Possible Paths

[10] For more information on the measures implemented as well as the research on lexical/thematic chaining and the performance of GLexi, the lexical chainer for German corpora, please refer to Cramer and Finthammer (2008) and Cramer et al. (accepted) respectively. See e.g. Jiang and Conrath (1997), Leacock and Chodorow (1998), Lin (1998), Resnik (1995), Wu and Palmer (1994) for more information on semantic measures drawing on word nets.

[11] The three Google measures are based on co-occurrence counts and use different algorithms to convert these counts into values representing semantic relatedness. See e.g. Cilibrasi and Vitanyi (2007) for more information on this.

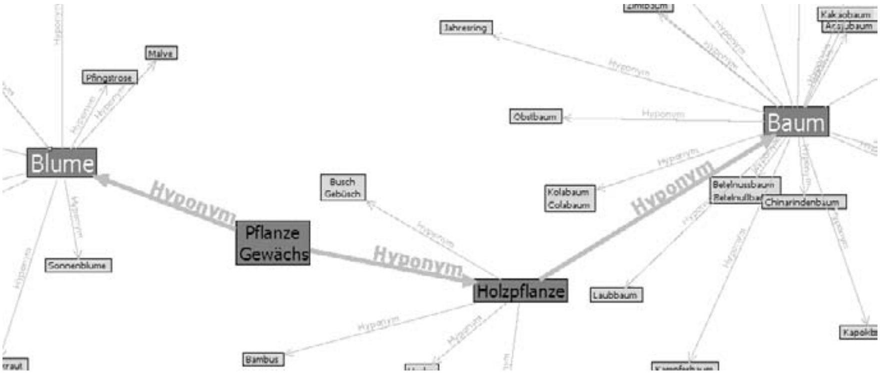
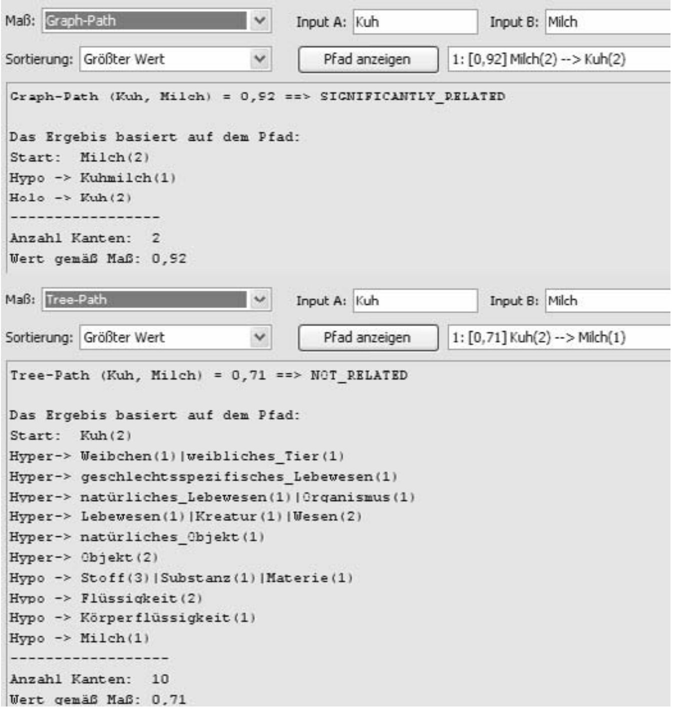


Figure 12. Screenshot GermaNet Pathfinder: Path Visualization Function

following features: In order to calculate the relatedness for a given word-pair or pair of synsets (see Figure 10), the user may select a single measure or all measures at one time. Furthermore, the relatedness values can be calculated with respect to all possible synset-synset combinations or one particular combination (see Figure 11). In order to analyze and compare the relatedness values of the different measures, the GermaNet Pathfinder includes a function to calculate the relatedness for a complete list of word-pairs or pairs of synsets, the results of which are stored in .cvs-format. We found that this function is especially useful for evaluating the performance of semantic measures with regard to a given task. Finally, the relatedness value, which corresponds – in the case of the GermaNet-based measures – to a path, can be examined visually using the corresponding GermaNet Pathfinder functions (see Figure 12).

Semantic relatedness measures play a key role in NLP and consequently represent central components of many applications. Therefore, CL students should be familiar with the basic concepts and algorithms of semantic relatedness. However, in order to understand and be able to independently and productively use semantic relatedness measures, theoretical and practical knowledge (or experience) is required. For this purpose a hands-on seminar might be most suitable. Using the Pathfinder, CL students may explore the various aspects of GermaNet considered in the calculation of semantic relatedness, such as path length, relation types, and graph depth. By comparing various paths between a given word pair, subtle differences between the algorithms can be discussed. An example is shown in Figure 13, which illustrates the difference between relatedness measures drawing on the complete GermaNet graph as a resource and those exclusively using the hyponym-tree. The GermaNet Explorer and Pathfinder may also help analytically retrace paths between semantically more or less related pairs of words or synsets. These manually constructed paths (which



Maß: Graph-Path Input A: Kuh Input B: Milch

Sortierung: Größter Wert Pfad anzeigen 1: [0,92] Milch(2) --> Kuh(2)

```
Graph-Path (Kuh, Milch) = 0,92 ==> SIGNIFICANTLY_RELATED

Das Ergebis basiert auf dem Pfad:
Start:  Milch(2)
Hypo -> Kuhmilch(1)
Holo -> Kuh(2)
-----------------
Anzahl Kanten:  2
Wert gemäß Maß: 0,92
```

Maß: Tree-Path Input A: Kuh Input B: Milch

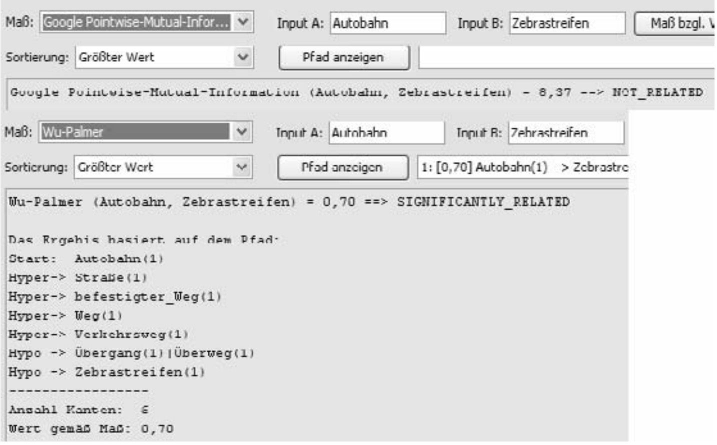
Sortierung: Größter Wert Pfad anzeigen 1: [0,71] Kuh(2) --> Milch(1)

```
Tree-Path (Kuh, Milch) = 0,71 ==> NOT_RELATED

Das Ergebis basiert auf dem Pfad:
Start:  Kuh(2)
Hyper-> Weibchen(1)|weibliches_Tier(1)
Hyper-> geschlechtsspezifisches_Lebewesen(1)
Hyper-> natürliches_Lebewesen(1)|Organismus(1)
Hyper-> Lebewesen(1)|Kreatur(1)|Wesen(2)
Hyper-> natürliches_Objekt(1)
Hyper-> Objekt(2)
Hypo -> Stoff(3)|Substanz(1)|Materie(1)
Hypo -> Flüssigkeit(2)
Hypo -> Körperflüssigkeit(1)
Hypo -> Milch(1)
------------------
Anzahl Kanten:  10
Wert gemäß Maß: 0,71
```

Figure 13. Path Based on Complete GermaNet-Graph (Graph-Path) vs. Path Based on Hyponym-Tree (Tree-Path)

in this case correspond to semantic relatedness values) can then be compared with the automatically calculated ones. This might raise the students' awareness of aspects in need of improvement: on the one hand in the modeling of GermaNet and on the other hand in the algorithms of the semantic relatedness measures. The example shown in Figure 13 (i.e. the path between `Kuh`, Engl. cow, and `Milch`, Engl. milk) demonstrates that some (if not most) paths do not reproduce human intuition and thus differ from the paths humans would select. Finally, the comparison of the three Google-based measures (relying on corpus statistics) and the eight GermaNet-based ones (relying on manually created structures) may clarify which aspects of the human intuition on semantic distance are included in the measures using different resources. Further, the comparison may highlight the ways in which the measures diverge, i.e. syntagmatic vs. paradigmatic relations or simply coverage.



Maß: Google Pointwise-Mutual-Infor...
Input A: Autobahn
Input B: Zebrastreifen
Sortierung: Größter Wert

```
Google Pointwise-Mutual-Information (Autobahn, Zebrastreifen) - 8,37 --> NOT_RELATED
```

Maß: Wu-Palmer
Input A: Autobahn
Input B: Zebrastreifen
Sortierung: Größter Wert
1: [0,70] Autobahn(1)   > Zebrastre

```
Wu-Palmer (Autobahn, Zebrastreifen) = 0,70 ==> SIGNIFICANTLY_RELATED

Das Ergebis basiert auf dem Pfad:
Start:  Autobahn(1)
Hyper-> Straße(1)
Hyper-> befestigter_Weg(1)
Hyper-> Weg(1)
Hyper-> Verkehrsweg(1)
Hypo -> Übergang(1)|Überweg(1)
Hypo -> Zebrastreifen(1)
-----------------
Anzahl Kanten:  6
Wert gemäß Maß: 0,70
```

Figure 14. Google-based vs. GermaNet-based Measure: Freeway – Crosswalk

4 Outlook

We plan to continue introducing word nets, lexical semantic concepts, and algorithms of semantic relatedness using the GermaNet Explorer and GermaNet Pathfinder. As mentioned above, we found that the tools might indeed support the learning process of our students. However, we think in order to successfully employ the two tools it will be necessary to carefully design, deploy, and evaluate seminar sessions. Therefore, we plan to employ both in our courses in an even more focused manner. In doing so, we intend to collect information on the following aspects:

- How may lessons on lexical semantics be enriched by using both tools?
- How may word nets, such as Princeton WordNet or GermaNet, be effectively introduced and presented?
- Which tasks or student projects respectively are suitable for accomplishing this objective?

In recent years, academic higher education teachers (particularly, but not only in computational linguistics) have dedicated a considerable amount of commitment to the development of courses. The exchange of ideas at conferences and workshops devoted to this topic[12] has disclosed many interesting experiences. We argue that it is worthwhile to write these up so that more teachers may benefit from these insights.

[12] See e.g. TeachCL-08 (`http://verbs.colorado.edu/teachCL-08/`)

Consquently, we plan to test our ideas for tasks as outlined in Sections 2 and 3 in our courses. When the first positive experiences can be validated, we intend to compile a teaching plan and make it publicly available[13].

References

Banerjee, Satanjeev and Ted Pedersen (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, 136–145, Springer.

Barzilay, Regina and Michael Elhadad (1997). Using Lexical Chains for Text Summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop*, 10–17.

Boyd-Graber, J., C. Fellbaum, D. Osherson, and R. Schapire (2006). Adding Dense, Weighted, Connections to WordNet. In *Proceedings of the 3rd Global WordNet Meeting*, 29–35.

Budanitsky, Alexander and Graeme Hirst (2006). Evaluating WordNet-Based Measures of Semantic Relatedness. *Computational Linguistics* 32 (1):13–47.

Carthy, Joe (2004). Lexical Chains versus Keywords for Topic Tracking. In *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, 507–510, Springer.

Cilibrasi, Rudi and Paul M. B. Vitanyi (2007). The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering* 19(3):370–383.

Cramer, Irene and Marc Finthammer (2008). An Evaluation Procedure for Word Net Based Lexical Chaining: Methods and Issues. In *Proceedings of the 4th Global WordNet Meeting*, 120–147.

Cramer, Irene, Marc Finthammer, Alexander Kurek, Lukas Sowa, Melina Wachtling, and Tobias Claas (accepted). Experiments on Lexical Chaining for German Corpora: Annotation, Extraction, and Application. *LDV-Forum* Ontologies and Semantic Lexical in Automated Discourse Analysis.

Fellbaum, Christiane (ed.) (1998). *WordNet. An Electronic Lexical Database*. The MIT Press.

Finthammer, Marc and Irene Cramer (2008). Exploring and Navigating: Tools for GermaNet. In *Proceedings of the 6th Language Resources and Evaluation Conference*.

Green, Stephen J. (1999). Building Hypertext Links By Computing Semantic Similarity. *IEEE Transactions on Knowledge and Data Engineering* 11(5):713–730.

Gurevych, Iryna and Hendrik Niederlich (2005). Accessing GermaNet Data and Computing Semantic Relatedness. In *Companion Volume of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005)*, 5–8.

Hearst, Marti A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th conference on Computational linguistics*, 539–545.

Hirst, Graeme and David St-Onge (1998). Lexical Chains as Representation of Context for the Detection and Correction Malapropisms. In Christiane Fellbaum (ed.), *WordNet: An Electronic Lexical Database*, 305–332, The MIT Press.

Jiang, Jay J. and David W. Conrath (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of ROCLING X*, 19–33.

Kunze, Claudia (2001). *Computerlinguistik und Sprachtechnologie: Eine Einführung*, chapter Lexikalisch-semantische Wortnetze, 386–393. Spektrum.

Leacock, Claudia and Martin Chodorow (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. In Christiane Fellbaum (ed.), *WordNet: An Electronic Lexical Database*, 265–284, The MIT Press.

[13] We will possibly also publish it on the ACL wiki, which provides a repository of teaching material (`http://aclweb.org/aclwiki/index.php?title=Teaching`).

Lin, Dekang (1998). An Information-Theoretic Definition of Similarity. In *Proceedings of the 15th International Conference on Machine Learning*, 296–304.

Mandala, Rila, Takenobu Tokunaga, and Hozumi Tanaka (1998). The Use of WordNet in Information Retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, 31–37.

Miller, George A. and Florentina Hristea (2006). WordNet Nouns: Classes and Instances. *Computational Linguistics* 32(1):1–3.

Novischi, Adrian and Dan Moldovan (2006). Question Answering with Lexical Chains Propagating Verb Arguments. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 897–904.

Patwardhan, Siddharth and Ted Pedersen (2006). Using WordNet-Based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *EACL Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together*, 1–8.

Resnik, Philip (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the IJCAI 1995*, 448–453.

Tanács, Attila, Dóra Csendes, Veronika Vincze, Christiane Fellbaum, and Piek Vossen (eds.) (2008). *Proceedings of the 4th Global WordNet Conference*, University of Szeged, Department of Informatics.

Vossen, Piek (ed.) (1998). *EuroWordNet: a Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers.

Wu, Zhibiao and Martha Palmer (1994). Verb Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 133–138.

Adaptive word sense views for the dictionary database eWDG: The case of definition assignment

Axel Herold and Alexander Geyken

Abstract. We describe an approach that uses explicit and implicit references in the dictionary database eWDG to automatically assign appropriate definitions to undefined headwords. The approach is developed and evaluated in the context of a lexical information system that aims to generate adaptive views on dictionary and corpus resources.

1 Introduction and project context

The approach described in our paper is being developed in the context of the dictionary project “Digitales Wörterbuch der deutschen Sprache” (DWDS, en.: “Digital Dictionary of the German Language”). This long-term project was launched in 2000 at the Berlin-Brandenburg Academy of Sciences (BBAW) with the goal to build an online lexical information system that provides access to multiple German text corpora and lexical resources (Klein 2004a,b; Klein and Geyken 2000). The current release of this information system (Geyken 2005) constructed in the first project phase from 2000 to 2006 offers flexible access to four different types of resources:[1] (1) a corpus component with currently 800 million word tokens in total (Geyken 2007), (2) a collocation component offering options to compute collocations for a lexical unit according to common statistical measures, (3) a thesaurus component that computes synonyms, hyponyms, and hypernyms for lexical units on the basis of the dictionary data (Geyken and Ludwig 2003), and (4) the dictionary database eWDG (electronic WDG). This database builds on a six-volume print dictionary of German, the “Wörterbuch der deutschen Gegenwartssprache” (WDG, en.: “Dictionary of Present-day German”) published between 1952 and 1977 and compiled at the Deutsche Akademie der Wissenschaften[2]. The print dictionary comprises over 4 500 pages and contains more than 60 000 headwords (more than 120 000 if compounds are counted separately). The term “deutsche Gegenwartssprache” (en.: “German present-day language”) is understood broadly by the lexicographers; the dictionary is not restricted to the language spoken and written in the middle of the

[1] `http//www.dwds.de`

[2] since 1972: Akademie der Wissenschaften der DDR

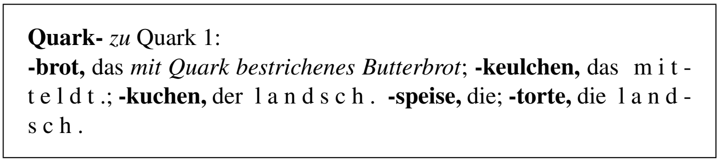
20th century, but also incorporates sources from the 18th and 19th centuries insofar as these are still widely read (Malige-Klappenbach 1986; Wiegand 1990).

Our paper concentrates on the current version of the eWDG dictionary database (Schmidt et al. in press) and presents an approach to generate word sense views on the eWDG entries. In section 2 we will briefly introduce the concept of a word sense view and explain the main problem that one has to deal with in this context. A considerable amount of headwords in the eWDG does not have definitions explicitly assigned to them in the dictionary entry in which they occur. However, for many of these headwords appropriate definitions exist in other dictionary entries, and these definitions can be detected by locating the targets of explicit and implicit cross-references. The main goal of our approach is to create and to evaluate strategies which automatically assign appropriate definitions to undefined headwords. In section 3 we will outline those features of the eWDG's entry structure that are relevant for our strategy, and we will differentiate four different cases of undefined headwords. Section 4 will describe how we use implicit and explicit references to automatically assign definitions to the headwords in these four cases. Through the use of examples we will illustrate to what extent this assignment of definitions to headwords can be performed automatically on the basis of our current structural annotation of the eWDG (Schmidt et al. in press). In sections 5 and 6 we will discuss and reflect upon the results of our approach.

2 Adaptive word sense views in the context of a lexical information system

One general guideline in designing our lexical information system is that it should be adaptive in the sense that we want to offer specialized views on our lexical and corpus data for different dictionary functions (e. g. text comprehension, text production, linguistic research), as well as specialized search mechanisms for different user groups (e. g. journalists, translators, linguistic researchers, computational linguists). The concept of an adaptive view is crucial for this guideline. A view is a combination of data items stored in our resources that is presented to the user on a display medium. Normally the display medium is a computer monitor, but it may also be a smaller display, e. g. an e-phone or a handheld computer or a sheet of paper with a printed version of a dictionary entry. Generating adaptive views on our resources means selecting those datasets that are relevant for a specific usage situation and present the data in a way that is best suited to a given display medium. The concept of views originates from database theory (e.g. Date 2003) and has been transferred to the context of hypertext systems (Hammwöhner 1997; Lenz and Storrer 2006), adaptive hypermedia (Brusilovsky 2001), and text technology (Schmidt 2005). In the current phase of our project we aim to generate multiple and adaptive views on one annotated dictionary resource, the eWDG. This is the first step towards our general goal, to generate views that select and combine data items from several dictionaries and text resources.

The approach described in this paper focuses on *sense views*, i. e. views that select the relevant items to understanding the meaning of the headwords in the eWDG. The basis for generating these sense views is the logical (i. e. content-oriented) annotation of the dictionary entry structure that has been generated in a semi-automated process described in (Schmidt et al. in press). In this context we had to cope with the problem that the eWDG contains a considerable amount of headwords (approximately 56 000, cf. table 1) that do not have explicit definitions. This high number of undefined headwords is not a shortcoming of our database and its annotation: the respective headwords already lack explicit definitions in the printed WDG. This may be illustrated with the entry in figure 1: the printed WDG groups entries to blocks with one main headword – in this example “Quark-” (en.: “cream cheese”) as the modifier component of compounds – and several subordinate headwords – in this example the compounds “Quarkbrot” (en.: “cream cheese sandwich”), “Quarkkeulchen” (en.: “cream cheese fritters”), “Quarkkuchen” (en.: “cream cheese cake”), “Quarkspeise” (a sweet desert made from cream cheese) and “Quarktorte” (en.: “cream cheese cake”). Other types of nested blocks are described in section 3. The crucial aspect here regarding our approach is that only one out of the five headwords in this entry is followed by an explicit definition, namely “Quarkbrot.” For the other undefined headwords, however, explicit and implicit references can be exploited to locate appropriate definitions stored in other parts of the dictionary. For example, the main headword “Quark-” contains an explicit reference to the first sense of “Quark,” so that the definition of this sense can be selected and displayed in the sense view of the “Quark-” compound entry. Likewise, the compound entry “Quarkkeulchen” contains the component “Keulchen,” which is the headword of its own entry containing a definition. The subordinate headword may thus be used as an implicit reference to the entry headed by the corresponding headword, and the definition found in this entry may be displayed in the sense view of the compound.



Quark- *zu* Quark 1:
-brot, das *mit Quark bestrichenes Butterbrot*; **-keulchen,** das mitteldt.; **-kuchen,** der landsch. **-speise,** die; **-torte,** die landsch.

Figure 1. Initial example: compounding with “Quark-.”

The omission of explicit definitions in these cases is typical for print dictionaries:

on the one hand, the designers of a print dictionary attempt to provide as many headwords as possible; on the other hand, they aim to economize the production costs. In this situation, lexicographers apply strategies for shortening by omitting, and referencing lexical information, strategies that are also referred to as text condensation methods in metalexicographic research (for an overview, see Wolski 1989). This compact way to present entries is suited if the goal is to present a *typographic view*[3] of a dictionary, i. e. on a printed page. However, it is not appropriate for the attempt to adapt the required amount of information to the particular requirements of a given application.

For example, it would not be feasible to present an entire entry block – for complex entries, an entry block contains well over 100 lines – for a handheld application when the specific information about the entry indeed only consists of three lines: morphological information, the definition, and optionally one or two citations. For this purpose, it is necessary to annotate the content of the eWDG entries explicitly; i. e. in this so-called *lexical view* in our dictionary, we want to encode all references to definitions explicitely in the entry structure.

3 Entry types and definitions in the WDG

Every entry of the WDG consists at least of a headword and is optionally followed by a selection of grammatical, etymological, phonological and usage information, and a definition as well as one or more senses. Senses in turn consist mainly of (made up) usage examples for the headword, literary citations, and, in the case of several distinct senses for a headword, they typically contain (additional) definitions as well.

For our purpose we differentiate between four different types of entries in the printed version of the WDG:

Main entries (`main`) are entries starting with a headword, specified in its complete orthographic form, and followed by further information (cf. “findig,” “Fin de Siècle,” “fünferlei,” and “funktionell” in figure 2). Only about 2 000 out of about 30 000 main entries occur without an explicit definition. Still, all of them possess either a “colloquial definition” or at least a reference to another `main` entry or a specific sense thereof.

Derivational entries (`deriv`) or lists thereof are always part of a nested entry block headed by a `main` entry that directly precedes it. In figure 3 “Liebhaberin” (en.: “(female) lover, faddist”), the feminine form is regularly derived from the word “Liebhaber,” which is the headword of the `main` entry in the nested group.

[3] `http://www.tei-c.org/release/doc/tei-p5-doc/html/DI.html#DIMV`

Every one of the 6 000 deriv entries is headed by an explicit reference to the main entry that represents the derivational base for it, possibly restricting the correspondence on a selected set of senses, as can be seen for “Liebhaberin” in figure 3.

Compositional entries type I (comp1) are used for compounds. They do not specify the complete form of the headwords, but only the head element of the compound. The left attachments, i. e. the modifier elements of the compounds, are only stated once at the beginning of the block that embeds the comp1 entry. Typically, the left attachment is identical to the headword of the preceding main entry, but there are 2 500 entries out of about 55 000 in which this is not the case. In these, an explicit pointer to a main entry is used to find the proper resolution to the reference (cf. figure 1 in which the left attachment “Quark-” is referenced with one sense of the corresponding main entry only).

Compositional entries type II (comp2, often called “petit compounds” with regard to the small font size used in the printed edition) are complementary to the comp1 entries: the head element of these compounds corresponds to the headword of the preceding main entry (“Liebhaber” in the example in figure 3), and a range of possible modifier elements is specified (“Hunde-, Pferde-, Vogelliebhaber”, etc. in our example). Semantically the set of comp2 entries for an entry *e* comprises different types or special kinds of *e*. About 81 000 comp2 entries were eventually included in the dictionary to exemplify the huge potential of compounding in German (Malige-Klappenbach 1986: 20 f.). While comp2 entries almost always only state the existence of their headword but give no further details as to grammatical or usage information, there is considerable overlap between comp1 and comp2: about 70 % of the comp2 entries also appear as comp1 entries that are provided with substantially more information, often including definitions. Still, about 25 000 headwords only appear as part of comp2 sets.

The alphabetic sequence of entries is overlaid by a second ordering principle that groups together entries into blocks according to processes of word formation. Every main entry starts a new block and is optionally followed by a list of deriv entries, which in turn is optionally followed by a list of comp2 entries as in figure 3. Also, comp1 entries are grouped together in blocks with a trailing optional set of comp2 entries. According to the nest-alphabetic ordering, it is only among the block-initial entries and within the sets of the different entry types that the sequence of headwords is ordered alphabetically.

If we bear in mind the systematic overlap between comp1 entries and the generally undefined comp2 entries, about 56 000 entries in the eWDG do not contain explicit definitions (cf. table 1). Put differently, almost every second entry in the

findig */Adj./ einfallsreich und schlau: …*

Fin de siècle, das; -, */ohne Pl./* … <franz.> */Bezeichnung für das dekadente bürgerliche Lebensgefühl am Ende des 19. Jahrhunderts und seinen Ausdruck in Literatur und Kunst/*

fünferlei, *vgl. dreierlei*

funktionell */Adj./* <lat.>
1. Wissensch. */entsprechend der Bedeutung 1 a* von *Funktion/ …*
2. Med. */entsprechend der Bedeutung 1 b α* von *Funktion/ …*

Figure 2. Examples of a `main` entries with different definition types, underlined are definitions and references (not underlined in the print version).

Liebhaber, der; -s, -
1. *Mann, der eine Frau liebt*: …
a) *oft* abwertend *Mann, der mit einer Frau ein Liebesverhältnis hat, Geliebter einer Frau*: …
b) veraltend *Verehrer einer Frau, Bewerber um eine Frau*: …
c) Theater …
2. *jmd., der für etw. eine Vorliebe hat, Freund, Verehrer von etw.*: …
3. *Amateur …*
zu 1 c. 2. 3 **Liebhaberin,** die; -, -nen
zu 2 */in Verbindung mit Tieren z. B./* Hunde-, Pferde-, Vogelliebhaber; */in Verbindung mit Blumen z. B./* Blumen-, Rosen-, Tulpenliebhaber; */ferner in/* Kunst-, Musik-, Naturliebhaber

Figure 3. Explicit denotation of related senses in a reference from `deriv` to `main`

eWDG, regardless of its type, does not contain an explicit definition. On the one hand, this situation is due to economic reasons (e. g. limitations of printing space); on the other hand, there might be instances of `comp1` and `comp2` that are fully transparent compounds in which the meanings of their parts are compositional. Despite that, there is a huge demand for explicitly defined entries as the eWDG proves to be an invaluable and frequently used resource, not only for the native speaker, but also for the language learner.

In the printed version of the WDG, proper definitions of headwords appear in

Table 1. Fraction of unique entries with vs. without explicit definition according to their type.

	`main`	`deriv`	`comp1`	`comp2`	total
defined	28 405	0	33 052	0	61 457
undefined	2 182	5 884	22 345	25 729	56 140
total	30 587	5 884	55 397	25 729	117 597

italics and before any sense distinctions within the entry but sometimes also at the beginning of sense descriptions. In the eWDG, all proper definitions are represented by `<def>` elements according to TEI P5. There is a total of 94 383 explicit definitions in the current eWDG.

General purpose labeling is another method employed for providing definitory information. Enclosed in slashes in the print version and marked up as `<lbl>` elements in the eWDG, general labels serve the purpose of providing information that is not easily expressed using any other annotation. Every type of lexicographic information can be found in these sections (Schmidt et al. in press). Consequently, `<lbl>` elements are difficult to parse. As the transformation process of the eWDG from typographic to content-oriented markup is currently not yet completely finalized, the fraction of definitory information in general purpose labels can only be estimated. The preliminary heuristic we used is based on the number of tokens within a `<lbl>` element. About 60 % of the general purpose labels contain only one token which typically denotes the part-of-speech of the headword. Another 15 % contain two tokens often expressing constraints with respect to the inflection of the headword (e. g. “ohne/nur Pl.,” en.: “without/only plural”). Accordingly, a reasonable estimate would be that less than 25 % of those labels contain definitory text. No more than about 2 000 undefined entries contain a `<lbl>` element that satisfies the token count criterion and thus might be defined by the content of the general label.

Entries may contain references to other entries. By far the most frequent referencing schema is based on entry relative position and type relations. This was discussed earlier in the description of the different entry types and the nest-alphabetic block ordering. If the deduction of an implicit reference would result in an unintended relation between entries by the lexicographer, an explicit pointer is consequently used to overwrite that reference. Also, any other reference between entries is expressed explicitly. In those cases the target indication can be as specific as addressing a – possibly deeply embedded – sense or a whole range of different senses.

Figure 2 gives some examples of definitions and references to other entries for different `main` entries. Both the most basic and most frequent type is represented by “findig” (“resourceful”), where the proper definition “einfallsreich und schlau” (en.: “inventive and clever”) is marked up in the printed version in an italics font and followed by a colon. So far, these definitions have been successfully transformed into the respective logical TEI P5 compliant markup (Schmidt et al. in press). In some cases, however, definitions are given in a more colloquial way, as for “Fin de siècle”. Here, a verbose explanation is noted as a general purpose label. The entry headed by “fünferlei” (en.: “five different kinds of ...”) is an example of an explicitly encoded reference to another `main` entry, namely *dreierlei* (en.: “three different kinds of ...”). The association is introduced by the functional abbreviation “vgl.” (en.: “compare”) and the headword to which the user is referred. The set of abbreviations used for referencing is finite and consists of only four members. Finally, “funktionell” is an example of an entry with explicit references to single senses as opposed to an overall entry.

4 Construction strategy

The reference system between entries as discussed in section 3 is summarized in figure 4. Our construction strategy for adaptive definition assignment is tightly bound to this reference system. Missing definititory information for entry e_n is inherited from the entry e_{n-x} that e_n is explicitly or implicitly referencing. For references to a distinct sense of e_{n-x}, the definition of that sense is used instead of a potential definition common to all senses of e_{n-x}.

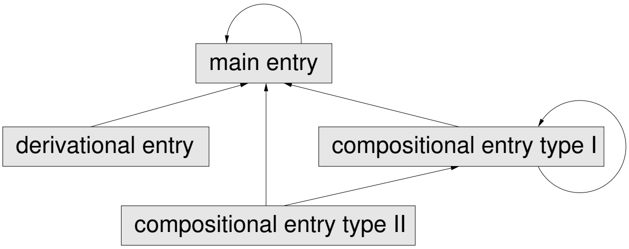


Figure 4. The complex referencing system between different entry types in the WDG.

Three general rules allow for the resolution of references from e_n to e_{n-x}:

1. Direct references are preferred over indirect references. If a definition is found in e_{n-x}, we do not resolve references originating from e_{n-x} to any other entries or senses. If a definition is still not found in e_{n-x}, the references of e_{n-x} are resolved unless no references originating from e_{n-x} are found. In this case, the process fails to locate a suitable definition. This rule is the break condition for the resolution process.

2. Explicit references are preferred over implicit references, as explicit references invalidate implicit references. This behavior was intended by the lexicographers in the construction of the WDG.

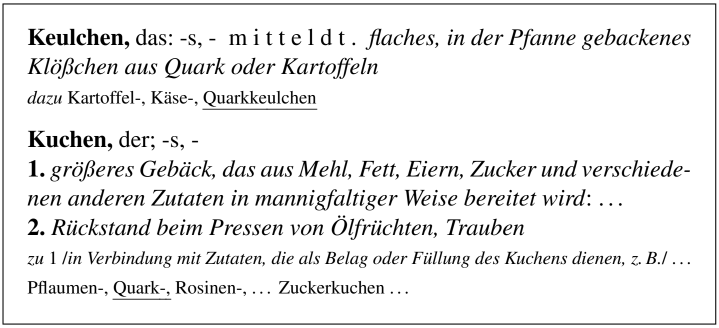
3. If e_n references e_{n-x} and e_{n-x} contains a headword orthographically identical to the headword of e_n, this reference is preferred over a reference to $e_{n-x'}$ where this is not the case.

In principle, the relation between entries is transitive, as figure 4 suggests. Therefore, the resolution process can be defined recursively. However, for reasons of usability and ease of use the target entry of a reference typically does not contain any explicit references in turn.

All senses and entries associated with e_n and located during the reference resolution are annotated within e_n, effectively making the relation between entries explicit by adding pointers. Basically, for this annotation we use TEI P5's `<ref>` elements embedded in `<xr>` elements to denote cross referencing phrases wherever appropriate (e. g. "vgl.," en.: "compare"). We do not copy any definitions found by the process directly into e_n, though, as this would result in a hypothesis – namely that the located definition was in fact appropriate information for e_n – being hard-coded into the dictionary database.

The different entry types described in section 3 are highly restricted in their utilization of explicit and implicit references. Every `main` entry was intended to be self-contained in the original version of the WDG. Therefore, only explicit references to other `main` entries can be found, and they occur only rarely. Entries of type `deriv` and `comp1` comprise both explicit and implicit references, with `deriv` showing a strong preference for implicit references, as it is directly contained in the block that is headed by its associated `main` entry target. Typical cases for explicit references of `deriv` entries concern the restriction to single senses of the `main` entry. An example is shown in figure 3, where "Liebhaberin" (en.: "(female) lover, fancier, faddist") is regularly derived from the generic form "Liebhaber," and both entries share only a subset of senses. Another trigger for explicit referencing in `deriv` and `comp1` entries is homograph selection.

Finally, the example in figure 1 introduced in section 2 shows a typical scenario in which `comp2` entries are used. "Quarkkeulchen" is listed twice in the eWDG, both as a `comp1` and a `comp2` entry; the same is true for "Quarktorte," "Quarkkuchen," and "Quarkbrot" (cf. figure 5 where some occurrences of `comp2` entries with "Quark-" are shown). The `comp2` occurrence of "Quarkkuchen" illustrates an explicit sense selection (sense 1 of "Kuchen," en.: "cake") which is common practice in the WDG. Additionally, the sequence of `comp2` entries associated with "Quarkkuchen" is supplemented by definitory information affecting all entries – denoting the left component of the resulting compounds as the main ingredient of the cake. Rule 3 accounts for the occurrence of entries with identical headwords. Still, there are about 26 000 comp2 entries (i. e. 32 %) for which no corresponding comp1 entry can be found in the dictionary (e. g. “Quarkspeise”).



Keulchen, das: -s, - mitteldt. *flaches, in der Pfanne gebackenes Klößchen aus Quark oder Kartoffeln*
dazu Kartoffel-, Käse-, Quarkkeulchen

Kuchen, der; -s, -
1. *größeres Gebäck, das aus Mehl, Fett, Eiern, Zucker und verschiedenen anderen Zutaten in mannigfaltiger Weise bereitet wird*: …
2. *Rückstand beim Pressen von Ölfrüchten, Trauben*
zu 1 */in Verbindung mit Zutaten, die als Belag oder Füllung des Kuchens dienen, z. B./* …
Pflaumen-, Quark-, Rosinen-, … Zuckerkuchen …

Figure 5. Example *Quark-*, continued. Typical occurences of comp2 *Quark*-entries (underlined).

5 Results

The construction strategy described in section 4 was applied to the current version of the eWDG. We found that 1 246 out of the 2 182 undefined headwords in main entries are linked to other main entries, i. e. about 57 % of those entries can be supplied with a definition, as the reference target contains a definition item in every such case (cf. table 2). For main entries there are no references to other entry types.

Table 2. Reference from main entries to other main entries

	main
undefined	2 182
explicit entry ref.	1 246
implicit reference	—
not resolveable	938

The majority of undefined entries was found among the comp2, comp1 and deriv entries. For deriv entries, about 80 % of the directly associated entries (i. e. main entries) contain a definition. Not surprisingly, there were no occurrences of indirect targets for deriv entries. As for compound entries (comp1 and comp2), the rate of defined direct targets was notably lower (below 70 %). However, as these entries frequently have indirect targets as well (comp1 ⟶ main, cf. figure 4) the resolution process often terminates sucessfully when scanning the indirect target for definition items. As a result, almost every undefined comp2 headword can be supplied with a definition. For comp1, less than 4 % of the undefined occurrences cannot be linked to a related and defined entry. The results are summarized in table 3.

Due to the ongoing refinement of the eWDG annotation, we still expect these figures to change slightly. In other words, we expect to improve our success rate for definition assignment when we disambiguate the WDG's general purpose labels and replace them with more content-oriented annotation labels.

Table 3. Reference from deriv, comp1 and comp2 entries

	deriv	comp1	comp2
undefined	5 884	22 345	25 729
direct target with def.	4 860	16 815	19 768
direct target without def.	1 024	5 530	5 961
indirect target without def.	—	879	136

6 Conclusion and discussion

The construction approach described in section 4 appears to be a reasonable way to assign definition items to undefined headwords of the eWDG. We are currently preparing a study that aims to evaluate quality and appropriateness of the automatically assigned definitions. We expect definitions derived for comp2 entries for which no corresponding comp1 entry exists to be problematic as can be shown with regard to "Quarkspeise". Here the head of the compoud ("Speise") cannot be reliably resolved automatically without further hints to which entry or sense thereof it relates ("Speise" appears as a homograph in the WDG). The definition assignment for the other undefined headwords in the "Quark"-entry is much more accurate because the entries for the head components of the compound can be located.

Another problem is that not all the cross-references in the original print dictionary are correct. In some cases comp2 entries were arbitrarily assigned to senses of main entries. This is due to the "traditional" creation process of the WDG; since the lexicographers worked in alphabetical order, they sometimes had to set cross-references to parts of the dictionary that were not yet completed (Malige-Klappenbach 1986: 20 f.). It is yet unclear how frequent this class of errors is and to what extent it will influence the resolution process. We will be able to quantify this in the study regarding the quality of our definition assignment.

In the framework of adaptive display of lexical information, the special case of comp1 and comp2 entries with identical headwords can probably be handled by merging items of the two entries. This might mean discarding the `comp2` entry completely from the sense view in many cases. However, as we illustrated in the example of "Quarktorte," `comp2` entries may also contain valuable sense information.

More definition items will be available when we complete the refinement and disambiguation of the eWDG. The completion of the remaining undefined headwords will then be subject to human lexicographic work. Human competence will also be needed to evaluate and improve the quality of the automatically assigned definitions. Still, the approach described in this paper supports lexicographers significantly in checking for coherent definitions across related entries and in assigning new definitions.

Finally, the rich cross-reference system that will be made explicit during the annotation refinement of the eWDG will allow one to generate user-specific word sense views that account for the complex associations between headwords encoded in the dictionary database.

References

Brusilovsky, Peter (2001). Adaptive Hypermedia. *User Modelling and User-Adapted Interaction* (11):87–110.

Date, Chris J. (2003). *An Introduction to Database Systems*. Amsterdam: Addisson Wesley, 8th edition.

Geyken, Alexander (2005). Das Wortinformationssystem des Digitalen Wörterbuchs der deutschen Sprache des 20. Jahrhunderts (DWDS). *BBAW Circular* (32).

Geyken, Alexander (2007). The DWDS corpus: A Reference Corpus for the German Language of the 20th Century. In Christiane Fellbaum (ed.), *Collocations and Idioms*, 23–40, London: Continuum Press.

Geyken, Alexander and Rainer Ludwig (2003). Halbautomatische Extraktion einer Hyperonymiehierarchie aus dem Wörterbuch der deutschen Gegenwartssprache. In *TaCoS 2003*, Gießen, 13.–15.6. 2003.

Hammwöhner, Rainer (1997). *Offene Hypertextsysteme. Das Konstanzer Hypertextsystem (KHS) im wissenschaftlichen und technischen Kontext*. Konstanz: UVK.

Hausmann, F. J., O. Reichmann, H. E Wiegand, and L. Zgusta (eds.) (1989, 1990). *Wörterbücher. Ein internationales Handbuch zur Lexikographie*. Berlin / New York: de Gruyter, 2 volumes.

Klein, Wolfgang (2004a). Das digitale Wörterbuch der deutschen Sprache des 20. Jahrhunderts. In Jürgen Scharnhorst (ed.), *Sprachkultur und Lexikographie*, 281–309, Frankfurt/M.: Peter Lang.

Klein, Wolfgang (2004b). Vom Wörterbuch zum digitalen lexikalischen System. *Zeitschrift für Literaturwissenschaft und Linguistik* (136).

Klein, Wolfgang and Alexander Geyken (2000). Projekt "Digitales Wörterbuch der deutschen Sprache des 20. Jahrhunderts". In *Jahrbuch der BBAW 1999*, 277–289, Berlin: Akademie-Verlag.

Lenz, Eva Anna and Angelika Storrer (2006). Generating Hypertext Views to Support Selective Reading. In *Proceedings of "Digital Humanities" 2006, Paris*, 320–323.

Malige-Klappenbach, Helene (1986). *Das Wörterbuch der deutschen Gegenwartssprache: Bericht, Dokumentation und Diskussion*. Tübingen: Niemeyer.

Schmidt, Thomas (2005). *Computergestützte Transkription. Modellierung und Visualisierung gesprochener Sprache mit texttechnologischen Mitteln*. Frankfurt/M.: Peter Lang.

Schmidt, Thomas, Alexander Geyken, and Angelika Storrer (in press). Refining and exploiting the structural markup of the eWDG. In *Proceedings of EURALEX 2008, 15–17 July 2008*, Barcelona.

Wiegand, Herbert Ernst (1990). Die deutsche Lexikographie der Gegenwart. In Hausmann et al. (1989, 1990), 2100–2246, 2nd volume.

Wolski, Werner (1989). Formen der Textverdichtung im allgemeinen einsprachigen Wörterbuch. In Hausmann et al. (1989, 1990), 1st volumne.

Research on dictionary use and the development of user-adapted views

Carolin Müller-Spitzer

Abstract. The development of user-adapted views of lexicographic data is frequently in demand by dictionary research on electronic reference works and hypertext information systems. In the printed dictionary it has been indispensable to develop a complete dictionary relative to a user group and using situations. In contrast, for any electronic presentation of lexicographic data there are possibilities to define user-specific views of an initially user-unspecific resource. However, research on the use of dictionaries in general, still has to answer several open questions as far as this subject is concerned. This paper will firstly provide an overview of the present state of research on dictionary use with respect to electronic lexicography. Subsequently, explanations of further prerequisites for a possible user-adapted access to data are followed, as exemplified by OWID, the Online Vocabulary Information System of the Institut für Deutsche Sprache. Finally, it will be outlined what results on the subject have been accomplished so far. Also the prospects of potential user-adapted presentations of lexicographic data will be highlighted.

1 The subject matter of this paper

Dictionaries are basic commodities or more precisely: “Der genuine Zweck eines Wörterbuchs besteht darin, daß es benutzt wird, um anhand lexikographischer Daten in den Teiltexten mit äußerer Zugriffsstruktur [...] Informationen zu denjenigen Eigenschaftsausprägungen bei sprachlichen Ausdrücken zu erschließen, die zum jeweiligen Wörterbuchgegenstandbereich gehören.”[1] (Wiegand 1998: 299) Therefore, it seems to be appropriate to compile dictionaries for specific user groups and particular using situations. This applies at least to printed dictionaries. In the field of electronic lexicography, the question of how to integrate a potential user conceptually must be addressed in a completely different way. Instead of adapting the lexicographic description to particular users and functions, one has to think about how far the user can define her/himself by a specific look-up situation as it occurs in a user-unspecific resource. It is possible in the digital dictionary to obtain information from the same data pool and to incorporate it in a suitable way typically relevant to a typical usage

[1] In English: The genuine purpose of a dictionary is to be used in order to obtain information on those properties of linguistic expressions which belong to the domain of the dictionary; this information can be accessed via lexicographic data in partial texts with an outer access structure.

context. It is therefore possible to adapt lexicographical descriptions to specific user needs and using situation. Instead of the static entry found in a printed dictionary in which all potential information is closely compressed, the user will receive a digital hypertext dictionary entry “on demand” with the information relevant to the current user situation (cf. Storrer 2001: 64f.). However, user-adapted dictionaries have only so far been developed as prototypes (cf. Thielen et al. 1998). The fact that it has just been about prototypes up to now, however, is not so much due to it not being known which classes of data are typically relevant for which usual look-up situations. The more likely reason is that “a context adapting presentation of lexical information requires a linguistically motivated and finely-grained modelling of lexicographical data.” (Storrer 2001: 65).

Hence, in the project OWID (*Online-Wortschatz-Informationssystem Deutsch*, online-vocabulary-information system of German)[2] of the *Institut für Deutsche Sprache* (IDS), the lexicographic data available has been modelled in a very fine grained way in order to create a basis for a user-adapted presentation of data. However, the actual habits and needs when using electronic dictionaries need to be critically analysed first in order to achieve this goal adequately. Research on dictionaries has comparatively well established the question which types of data are predominantly used by which types of users in which using situations. However, such analyses have been performed on the basis of printed dictionaries for the most part. For a printed dictionary, it is sufficient to define the types of information and to include general guidelines for its layout. As far as a user-adapted presentation of a general lexicographic database in hypertextual environment is concerned, completely new questions have to be addressed which are only relevant for the electronic lexicography, i.e. general questions like: how do users navigate in electronic dictionaries? How do they use offered search options? When do clusters of specific word information stop to be user-friendly and when does unclearness set in?[3] More specifically we need to ask: should a user (i.e. while using a dictionary) create a personal profile at the beginning of a session (e.g. user type: non-native speaker, situation of use: reception of a text) and should s/he navigate in all entries with this profile? Or is it more user-friendly to be able to change one’s profile and to look at the same entry with different profiles? Or is it better to let the user her/himself determine whether s/he wants to look at information on the meaning or on grammar instead of deriving a specific display of different kinds of information from her/his individual profile? Likewise, being able to provide the information with explicit headlines, which was not possible in a printed dictionary for lack of space, involves new challenges. For example, which types of terminology accommodate which user group the most has

[2] See `http://www.owid.de`.

[3] In addition, very general guidelines for the design of electronic media can be included, particularly web pages, but also special guidelines for online-dictionaries.(Cf. e.g. Almind 2005)

to be examined. All these questions show that research on use still has to answer many questions, especially for electronic dictionaries. Thus, in the following part, the present state of research on dictionary use – mainly with regard to electronic lexicography – will be outlined and some research methods applicable to the study of the use of electronic dictionaries will be described. In the third paragraph, OWID, an online dictionary portal, will be presented. On the one hand, the focus lies on the development of modelling as the basis of a user-adapted presentation and on the other hand on studies on the use of OWID which have already been accomplished. The paper finally concludes with an outlook on potential user-adapted presentations of lexicographic data in OWID.

2 On the present state of research on dictionary use

User research represents the main area of lexicographical research (cf. Wang 2001: 53). This statement might certainly overvalue the status of research on dictionary use as part of general meta-lexicographic research even though the situation has clearly improved over the past two decades: 25 years ago, one could say of the dictionary user: everyone talks about him, but no one knows him. Wiegand called him the unknown of the unknown. Today, one cannot speak of such a lexicographical “Yeti”. Not everything is known about him, but certainly more is now known (cf. Almind and Bergenholtz 2000: 259). In addition to research on the dictionary use of printed dictionaries (cf. i.e. Atkins 1998; Atkins and Varantola 1998; Ripfel 1989; Wiegand 1998), the use of electronic dictionaries has been increasingly examined. Infact, Engelberg and Lemnitzer still wrote in 2001 that there are hardly any studies about how the user’s behaviour is affected by innovations in the field of electronic lexicography (cf. Engelberg and Lemnitzer 2001: 71). However, in the meantime, this situation has improved to a small extent. Nevertheless, it is still true that research on use requires an enormous investment which seems to be justifiable only within the scope of academic research. (Cf. Lemnitzer 2001: 247)[4] However, the users’ reactions have some influence on new editions of commercial dictionaries, insofar as they are reflected in the sales figures, the requirements of the book trade’s agents and in the incoming language queries (cf. Höhne 1991; Müller 1991). Academic dictionaries are not subject to this pressure to the same extent. For both academic as well as commercial lexicography, however, extensive research on the use is certainly a profitable undertaking.

Current research on use in the field of electronic dictionaries especially focuses on new opportunities of observing the behaviour of users in the internet, namely

[4] For example, the study presented in Atkins (1998) was only possible by fundings gained from EURALEX and AILA.

through the analysis of log-files for electronic dictionaries; the reason being that this method can be accomplished with manageable investment. However, there is only a small number of projects reporting on how they put this opportunity into practice. "Although the proposal to draw upon log files in order to improve dictionaries was already expressed in the mid-1980s [...], and although numerous researchers have reiterated this idea in recent years [...], very few reports have been published of real world dictionaries actually making use of this strategy." (Schryver and Joffe 2004: 187) Exceptions are the study presented by Lemnitzer (2001) and the *Concept of Simultaneous Feedback* (i.e. Schryver and Prinsloo 2000) which was used, for example, for the online-dictionary "Sesotho sa Leboa Dictionary Project (SeDiPro)"[5]. However, more and more online dictionary projects make use of the analysis of log files like the last Euralex conference in Barcelona has shown (cf. i.e. Měchura 2008).

For both studies and methods respectively, it becomes obvious that research on use is especially interesting when the project examined can be improved immediately based on the results. This was much more difficult in the case of printed dictionaries as they can only be investigated when they have been completely published. More and more Internet dictionaries are not published once but in following parts so that insights into the user's behaviour can already be included in the working process. As De Schryver critically points out in his work "Simultaneous Feedback (SF)": "Feedback from the envisaged target user group is systematically and continuously obtained while compilation is still in progress." (Schryver and Joffe 2004: 187) Their idea is that a dictionary can be seen as a service for a "community", and therefore, users have the opportunity to point out missing entries which should then be added immediately. The concept of SF implies two elements: on the one hand the analysis of log-files which takes place manually even if perspectively this is intended differently: "ultimately, the idea is that an automated analysis of the log files will enable the dictionary to tailor itself to each and every particular user. At present, the analysis of the log files is still largely done manually, in part with the aim to draw up typical user profiles that will then be fed into the projected adaptive and intelligent dictionary of the future." (`http://tshwanedje.com/sf/`, 22.04.2008) On the other hand, the second element is the request to the users to react via e-mail, a procedure which produces good results according to De Schryver and Prinsloo. "On the whole, one observes a very good correspondence between the formal feedback through the online feedback forms, and the informal feedback obtained by means of an analysis of the log files. This is a satisfying sign indeed and indicates that modifying and adapting dictionary contents based on log-stat trends is a feasible strategy." (Schryver and Prinsloo 2000: 194) However, it is believed here that the results gained through examples of feedback and the incoming number of reactions via e-mail are

[5] See `http://africanlanguages.com/sdp/`.

not entirely convincing to be considered as relevant enough a method of research on use. Also Lemnitzer reports that the request to react via e-mail was hardly used and that a third of all incoming e-mails consisted of letters with irrelevant information and personal complaints (cf. Lemnitzer 2001: 248).

The method of SF as well as the analysis of log-files in general are not necessarily methods of research on use, but initially just forms of receiving feedback concerning a particular dictionary with the goal of improving this dictionary. However, by generalising this log file and SF-data, results can be gained that can generally be further used for research on use. The question that arises here is which of the questions asked at the beginning can be answered through the analysis of log-files? The log files give rough information about the navigation of users such as: which search box did they use? Where did they click next? There is no information – depending on the technical composition of the page – about further navigation within the entry. The log-files also provide information about the following: what did the users search for? How did they spell the search word? This means that from the analysis of log-files, some indications for improving the search options can be derived (mainly from the analysis of unsuccessful search results) as well as indications for potential lacks of words in the dictionary. In the same way, ideas for improving the guidelines to the search functions can be derived from the analysis of unsuccessful searchings. In this context, a useful recommendation of Lemnitzer is that incorrect searches are not the mistake of the user, but an inadequacy of the user surface (cf. Lemnitzer 2001: 248). However, what cannot be interpreted from log-files are questions like: why are users not able to find a particular piece of linguistic information? Why did they actually search for a particular word? What information from the dictionary entry do they understand and what do they not? Generally speaking: the analysis of log-files is one of many alternatives in research on dictionary use. It should be used for any online-dictionary, but it does not replace other forms to examine dictionary use. Research on dictionary use in general should consist of different methods like written interview, observation, records and tests (cf. Wang 2001: 67; Albert and Koster 2002; Atkins and Varantola 1998).

What also has to be pointed out is that both of the dictionaries examined by Lemnitzer and De Schryver have a very flat microstructure. Lemnitzer himself admits that altogether, the results of the study are limited, which, among other things, is due to the flat structure of the dictionary data (cf. Lemnitzer 2001: 258). That is also a reason why the results are only partly applicable to dictionaries or lexicographic information systems respectively which have a much more elaborated and complex microstructure, such as *elexiko|(*, a dictionary of contemporary German in OWID.

3 OWID – an example

OWID, a project of the IDS, is a lexicographic Internet portal which is currently being compiled and contains lexicographic-lexicological reference works of the IDS; perspectively, it is planned that data from external projects is added. Originally, OWID has its roots based in the IDS project *elexiko* (cf. Klosa 2008)[6].

3.1 Outline of content

The main emphasis of OWID is on the presentation of corpus-based, academic lexicographical-lexicological works. The following dictionaries have been included in OWID:

- *elexiko*: This electronic dictionary consists of an index of about 300.000 short entries with information on spelling, syllabification, and inflection. In the near future, further information (e.g. on word formation) and corpus samples will be added for all lexemes. Furthermore, *elexiko* comprises over 900 fully elaborated entries of headwords which are highly frequent in the underlying corpus. These contain extensive semantic-pragmatic descriptions of lexical items in actual language use. The dictionary is being extended continuously by further elaborated entries (cf. Haß 2005; Klosa et al. 2006).

- *Neologismenwörterbuch* (Dictionary of Neologisms): This electronic dictionary describes about 800 neologisms of the 1990s (cf. Herberg et al. 2004). Neologisms of the 21st century will be added in the near future.

- *Diskurswörterbuch 1945-55* (Discourse Dictionary 1945-55): This dictionary is a reference work resulting from a larger study of lexemes that establish the notional area of “guilt” in the early post-war era in Germany (1945-55). It summarises the lexical and semantical results of this study in the form of about 80 lexical entries. (Cf. Kämper 2005)

Besides the above mentioned, the contents of the existing printed reference works “Handbuch Deutscher Kommunikationsverben” (Handbook of German Communication Verbs; Harras et al. 2004) as well as the “VALBU – Valenzwörterbuch deutscher Verben” (Valency Dictionary of German Verbs; Schumacher et al. 2004) will be revised and published electronically in OWID by spring next year.[7]

[6] Until recently, the *elexiko*-project held a double function: on the one hand it was the lexicographic portal of the IDS, on the other hand a corpus-based dictionary of modern German is simultaneously being compiled in *elexiko*. In order to better differentiate between the overall portal and the dictionary *elexiko* itself, an independent portal-project named OWID was founded last year.

The consolidation of different lexicographic works under the same technological roof, brings about challenges of a very different kind. Various questions arise in the process: to what extent can and should the contents of each project be coordinated? How are the contents modelled and structured? How can common access structures to data be developed? The different works shown in OWID are independent projects as far as lexicographic content is concerned. Nevertheless, for potential users it is essential to develop as much common access structures on the particular contents as possible, which means to present more than a random collection of unrelated dictionaries and lexicographic resources, respectively. Therefore, it was necessary to maintain the independence of each individual dictionary project, while, at the same time, ensuring the integration of all the different data. One goal has always been the modelling of the data in such a way that a user-adapted presentation should be possible without changing the database.

3.2 Guidelines of data modelling

A lot of time has been invested on the data-modelling-level which represents the basis of all further possible data uses, by modelling the micro and content structures of each dictionary respectively according to one standardised concept so that the same phenomena are structured in the same way and that all the data is finely-grained structured such that one can access it targetted and specifically. On the basis of a new concept for specifically tailor-made modelling of lexicographic data (cf. Müller-Spitzer 2006, 2007a,b,c), a complex XML-DTD-Library was developed. Each individual information unit is tagged individually and named after its genuine purpose in terms of content. Thus, data modelling for OWID approaches the kind of modelling demanded by Storrer (2001: 61f.). Unlike leXeML, a suggestion for an XML-based tag-language for lexicographic data by Geeb (2001) aiming at the integration of the user in the modelling process, the lexicographic data here is initially modelled independently from potential situations of use in a way that they can be extracted both adaptively and flexibly from the database. This type of data modelling has been considered an innovative approach in general lexicographic practice by Kunze and Lemnitzer (2007: 85ff.) as well as Schlaps (2007).

We decided to use a specifically tailor-made modelling because the XML-structure also serves as a model for compiling the lexicographic entries in the XML-Editor. So the more customised with respect to the particular lexicographic project the XML-structure is, the less a lexicographer needs an additional manual for editing the entry structure. However, one could easily transform this specifically tailor-

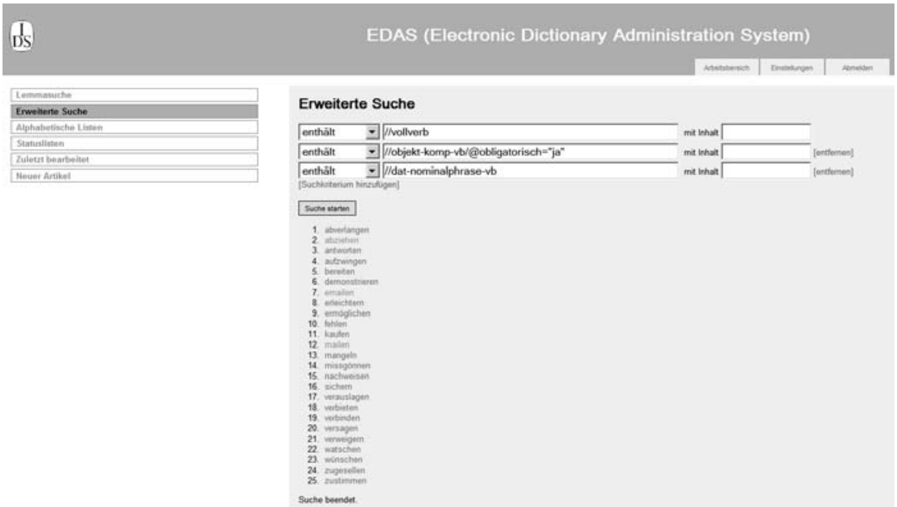
[7] VALBU will also be accessible via *grammis – the grammatical information system of the IDS*. (`http://hypermedia.ids-mannheim.de/grammis/`).

made structure into a standard one (like LMF or TEI). The XML detail of the entry "emailen" from the Dictionary of Neologisms shown in table 1 illustrates the tagging of information on valency and is an example of the overall granularity of tagging.

Table 1. XML detail of the entry "emailen"

```
<vb-valenz-neu>
<satzbauplan>
<satzbauplanA>jemand emailt (jemandem) (etwas)</satzbauplanA>
</satzbauplan>
<satzbauplan>
<satzbauplanA> jemand emailt (etwas) an jemanden</satzbauplanA>
</satzbauplan>
<satzbauplan>
<satzbauplanA>jemand emailt, dass [...]</satzbauplanA>
</satzbauplan>
<vb-komplemente-neu>
<subjekt-komp-neu obligatorisch="ja">
<nom-nominalphrase-neu/>
</subjekt-komp-neu>
<objekt-komp-vb obligatorisch="nein">
<dat-nominalphrase-vb/>
</objekt-komp-vb>
<objekt-komp-vb obligatorisch="nein">
<akk-nominalphrase-vb/>
<dass-satz-vb/>
</objekt-komp-vb>
<objekt-komp-vb obligatorisch="ja">
<praepositionalphrase-vb praeposition="an"/>
</objekt-komp-vb>
</vb-komplemente-neu>
</vb-valenz-neu>
```

In our internal editorial system, lexicographers are able to use this structure for advanced searches (with XPath expressions). For example, one can search for all regular verbs (`//vollverb`) which have obligatory object complements (`//objekt-komp-vb/@obligatorisch="ja"`), realised as a dative NP (`//dat-nominalphrase-vb`). In this example, the search results are entries from the *elexiko*- as well as from the neologism-dictionary (cf. fig. 1). It is one of *elexiko*'s aims to provide these extended search options for users too.[8]



Erweiterte Suche

enthält //vollverb mit Inhalt
enthält //objekt-komp-vb/@obligatorisch="ja" mit Inhalt
enthält //dat-nominalphrase-vb mit Inhalt

1. abverlangen
2. abziehen
3. antworten
4. aufzwingen
5. bereiten
6. demonstrieren
7. emailen
8. erleichtern
9. ermöglichen
10. fehlen
11. kaufen
12. mailen
13. mangeln
14. missgönnen
15. nachweisen
16. sichern
17. verauslagen
18. verbieten
19. verbinden
20. versagen
21. verweigern
22. watschen
23. wünschen
24. zugesellen
25. zustimmen

Suche beendet.

Figure 1. Advanced search options for lexicographers

The prerequisites for a user-adapted presentation of the data are already created on the database level. However, this capability is not yet visible on the present user surface of OWID. What is already offered is access to the data of *elexiko* as well as to the Dictionary of Neologisms through “Extended Search Options”. However, these could also undergo further differentiation and enlargement. For further improvements of OWID, we want to know more precisely how users access and navigate through the portal and how such a user-adapted presentation of data could be well arranged. So far, we accomplished first user observations on it which shall be briefly described in the following.

3.3 Pilot studies on using OWID and Elexiko

Although analyses of Internet log-files have been carried out in OWID, their evaluation does not cover studies over a larger period of time. Primarily, they show us what users enter into the search box. Except freak values like “Wie+mache+ich+einen+Hühnersalat” or “Wie+lange+braucht+man+um+die+Erde+zu+umkreisen”[9], log-

[8] The development of the Electronic Dictionary Administration System (cf. fig. 1) is being developed by Roman Schneider, a researcher of the IDS.

[9] In English: “How do I make a chicken salad”; “How long does it take to circle the earth”.

files show that most users search for single words or parts of larger expressions. Searchings with Boolean operators like "*lait*" are documented repeatedly, but they do not establish the majority of searches. This gives rise to the suspicion that the opportunity to search with Boolean operators should be explained at a more prominent place. Both, Scherer's study (2008) as well as a short study at the University of Mannheim support this hypothesis which is briefly described below. Furthermore, it becomes apparent that users look up newer words like "Patchworkfamily" or proper names of e.g. music bands like "Silbermond". Besides, there is a multitude of inconspicuous searches which cannot be assigned to a particular category. However, in searches like the orthographically aberrant "ddivergenzstellung" or "standart", it becomes apparent that the search methods must be more tolerant to mistakes, just as search entries like "plumpen", an inflected form of the adjective *plump*, show that ways of lemmatisation should be lodged in the search. Admittedly, the findings last mentioned are not new and were known before analysing log-files. However, these analyses show, for example, how many users look up information via the search box and who navigates via the list of headwords. Again, this is a very interesting information since the question whether a presentation of such a lemma-list is useful at all has been addressed repeatedly.

What cannot be concluded from log-files (in OWID) is which parts of the dictionary entry the user is looking at since this presentation is very interlaced. This is an important difference to the online-dictionary SeDiPro presented by De Schryver. Here, for example the entry "wish" consists of six equivalents with few and brief additional specifications, whereas in *elexiko*, the entry "wünschen" – as documented in printed pages – covers about 10 pages. Through log files it is possible to see that the user accesses a specific entry, but unfortunately (in our system) it cannot be seen whether the user navigates through specific senses and how s/he anticipates the given information. Through the analysis of the log-files alone in dictionaries like *elexiko*, it is not possible to obtain an insight into how a user navigates exactly within the information system. This illustrates the limits of a user observation of this kind: if one has a very flat-structured dictionary with little information on the search word, log-files pass on extensive information relative to the dictionary on to the lexicographer. If the microstructure is much more comprehensive, the analysis of log-files only provides limited results relative to the lexicographic product.

Furthermore, what is *not* searched for in OWID can be read from log-files. Interestingly, these are, for example, search words which indicate the demand for encyclopaedic information, which means there are hardly any searches for names of towns, for particular people etc. Presumably users seem to be aware of the fact that OWID is a linguistic reference book. Contrary to other studies (Lemnitzer 2001), hardly any searches for words from the so-called sexual and faecal vocabulary are documented. This might allow the assumption that OWID-users are not ordinary Internet users but people with more specific linguistic interests.

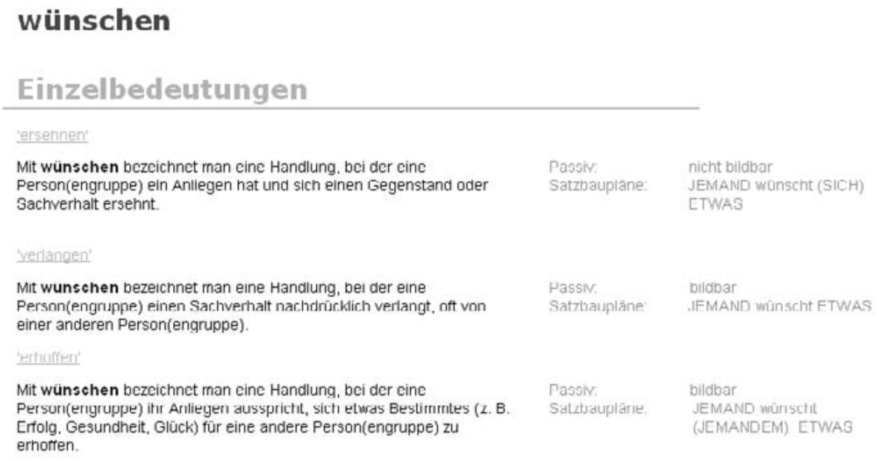
Scherer's study comprises a standardised online survey on OWID focussing on *elexiko*. 58 people were questioned through a standardised online-questionnaire which was constructed according to the multiple-choice method. A single session lasted 45-60 minutes. The study filled an existing gap in terms of research on users of electronic dictionaries – albeit only in the form of a pilot study so far. As Scherer points out, online studies in the form of standardised surveys which ask for subjective perceptions of the layout and functionality of a dictionary as well as the user's own evaluation of the applicability of the dictionary, have still not been carried out to a large extent (cf. Scherer 2008: 95). The use of standardised questionnaires is primarily interesting since it allows the levying of a great number of test people, whereas observation studies and records can only take small numbers of test people into consideration, and its analysis is also much more expensive. At the same time, the results of standardised interrogations frequently include information one would like to clarify further by asking questions in an interview. For example, one statement to be assessed in Scherer's study (translated):

> It is clear that the headword-list shown presents entries of all the four dictionaries.
> Strongly disagree 17%
> Disagree 37.7%
> Agree 39.6%
> Strongly agree 5.7%
> No statement 0%

In this case, it would be very informative to ask what is regarded unclear in the current presentation. (At the moment, the headwords of the different dictionaries are presented in different colours in OWID.) The result mentioned above is particularly difficult to assess in relation to the result of the next question:

> Do you remember what the design of different colours of the headwords means?
> It categorises headwords by part of speech 5.7%
> It categorises headwords by morphology 1.9%
> It categorises headwords by the dictionaries available in OWID 64.2%
> I don't know 24.5%
> No statement 3.8%

Another smaller study on the use of OWID and *elexiko* respectively was constructed as an observation study/interview: Within the scope of a seminar on "electronic lexicography" at the University of Mannheim, students were asked to look for five users who had to look up specific information in OWID/*elexiko* according to a questionnaire. Altogether there were about 60 test people. One session lasted about 30-45 minutes. The students should not only record all the using-actions, but also ask further questions and note down the answers to them. Again, we were able



wünschen

Einzelbedeutungen

'ersehnen'

Mit **wünschen** bezeichnet man eine Handlung, bei der eine Person(engruppe) ein Anliegen hat und sich einen Gegenstand oder Sachverhalt ersehnt.	Passiv: Satzbaupläne:	nicht bildbar JEMAND wünscht (SICH) ETWAS

'verlangen'

Mit **wunschen** bezeichnet man eine Handlung, bei der eine Person(engruppe) einen Sachverhalt nachdrücklich verlangt, oft von einer anderen Person(engruppe).	Passiv: Satzbaupläne:	bildbar JEMAND wünscht ETWAS

'erhoffen'

Mit **wünschen** bezeichnet man eine Handlung, bei der eine Person(engruppe) ihr Anliegen ausspricht, sich etwas Bestimmtes (z. B. Erfolg, Gesundheit, Glück) für eine andere Person(engruppe) zu erhoffen.	Passiv: Satzbaupläne:	bildbar JEMAND wünscht (JEMANDEM) ETWAS

Figure 2. Draft of a possible user-adapted online presentation of the *elexiko*-entry “wünschen”

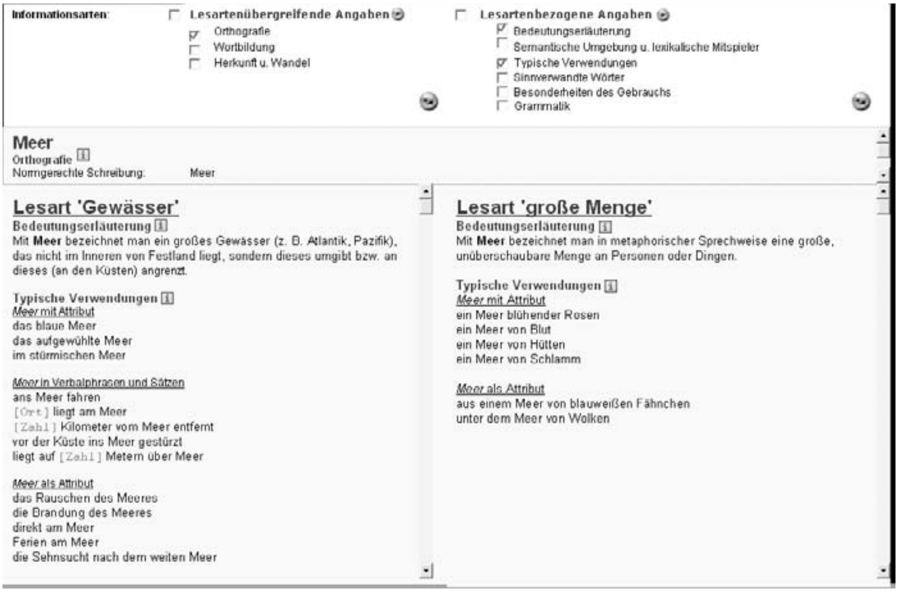
to learn from these tests. For example, the laypeople questioned (e.g. the students’ sisters and brothers or parents) had difficulties in particular with the terminology on the user surface. At “lesartenbezogene Angaben”[10], many assumed they would find information on pronunciation, for example. Besides, for some it was difficult to understand the fact that in *elexiko*, different batches of the lemma list are provided with very different information (i.e. that only some entries are fully elaborated).

So far, all of these pilot studies have not yet been backed up empirically in an adequate way in order to draw conclusions about the further development of user-adapted presentations. However, they give us a first impression about which field we should do further research in and towards where further developments should be directed to. Also important for OWID: the portal is currently being compiled which means the findings of the research on use can be included in the ongoing development of OWID.

3.4 Future prospects

Obviously, there is still a lot research work to be done regarding the use of electronic dictionaries. However, as said before, the question of what kind of information a dictionary user is searching for in a particular using situation is comparatively well

[10] In English: Information related to particular “readings” (senses) of the headword.



Informationsarten:

□ **Lesartenübergreifende Angaben**
☑ Orthografie
□ Wortbildung
□ Herkunft u. Wandel

□ **Lesartenbezogene Angaben**
☑ Bedeutungserläuterung
□ Semantische Umgebung u. lexikalische Mitspieler
☑ Typische Verwendungen
□ Sinnverwandte Wörter
□ Besonderheiten des Gebrauchs
□ Grammatik

Meer
Orthografie [i]
Normgerechte Schreibung: Meer

Lesart 'Gewässer'
Bedeutungserläuterung [i]
Mit **Meer** bezeichnet man ein großes Gewässer (z. B. Atlantik, Pazifik), das nicht im Inneren von Festland liegt, sondern dieses umgibt bzw. an dieses (an den Küsten) angrenzt.

Typische Verwendungen [i]
Meer mit Attribut
das blaue Meer
das aufgewühlte Meer
im stürmischen Meer

Meer in Verbalphrasen und Sätzen
ans Meer fahren
[Ort] liegt am Meer
[Zahl] Kilometer vom Meer entfernt
vor der Küste ins Meer gestürzt
liegt auf [Zahl] Metern über Meer

Meer als Attribut
das Rauschen des Meeres
die Brandung des Meeres
direkt am Meer
Ferien am Meer
die Sehnsucht nach dem weiten Meer

Lesart 'große Menge'
Bedeutungserläuterung [i]
Mit **Meer** bezeichnet man in metaphorischer Sprechweise eine große, unüberschaubare Menge an Personen oder Dingen.

Typische Verwendungen [i]
Meer mit Attribut
ein Meer blühender Rosen
ein Meer von Blut
ein Meer von Hütten
ein Meer von Schlamm

Meer als Attribut
aus einem Meer von blauweißen Fähnchen
unter dem Meer von Wolken

Figure 3. Draft of a possible online view of *elexiko* with an information display for customising the microstructure dynamically (entry "Meer")

investigated. Perspectively, this gives us the opportunity to outline potential user-adapted presentations in the near future, using *elexiko* as an example. It is important that these examples do not require any changes of the present database, but would be made possible solely by defining separate presentations through different XSLT-Stylesheets.

The entry "wünschen" from *elexiko* should serve as an example. In the current online-presentation, information on spelling and word formation is given on the first screen of the window that opens up by default and which summarises sense-unrelated details of a lexeme. Underneath, sense-related information is reached by clicking on links that represent individual senses.[11] Among this sense-related information, there is detailed information on the meaning, the semantic environment, on typical syntagmatic patterns, on sense-related paradigmatic words, such as synonyms and antonyms, on pragmatic and discursive features, and on grammar. One advantage of a user-adapted presentation certainly is that a potential user could create an individual profile at the beginning of an *elexiko* session. Supposing a non-native speaker of German in a situation of text reception requires an overview which is as brief as

[11] See `http://www.owid.de/pls/db/p4_anzeige.artikel?v_id=139627`.

possible including information on grammar in cases where there is a difference in meaning. Taking "wünschen" as an example, the grammatical information on the single senses differs. In such a situation of use, any other information could be left out, so that the most important information can be received at first sight (cf. fig. 2).

Another possibility would be to use a separate part of the screen for a kind of form where specific types of information can be ticked in boxes in order to be seen in the frame below. A user could appoint, for example, that for *elexiko* only the meaning definition and typical syntagmatic patterns are shown. That is probably closest to the idea of individually customised microstructures (cf. Engelberg and Lemnitzer 2001: 224 and Schryver 2003: 178ff. and see a first draft in figure 3). However, one would have to find out through empirical research which types of users are really aware of what kind of information is most useful in each individual situation of use.

The challenge of transforming these very novel presentation alternatives adequately alongside existing essential user needs for an online vocabulary information system certainly remains one of the most important and most interesting challenges in the field of electronic lexicography and research on dictionaries. And it is academic lexicography that should play a key role in mastering this task. Or, like Atkins says: "If new methods of access (breaking the iron grip of the alphabet) and a hypertext approach to the data stored in the dictionary do not result in a product light years away from the printed dictionary, then we are evading the responsibilities of our profession." (Atkins 1992, 521; cited from Schryver 2003: 144)

References

Albert, Ruth and Cor. J. Koster (2002). *Empirie in Linguistik und Sprachlehrforschung. Ein methodologisches Arbeitsbuch*. Tübingen.

Almind, Richard (2005). Designing internet dictionaries. *Hermes* 34:37–54.

Almind, Richard and Henning Bergenholtz (2000). Die ästhetische dimension der lexikographie. In Ulla Fix and Hans Wellmann (eds.), *Bild im Text – Text und Bild*, 259–288, Heidelberg.

Atkins, B. T. Sue (ed.) (1998). *Using Dictonaries. Studies of Dictionary Use by Language Learners and Translators*. Tübingen: Max Niemeyer.

Atkins, B. T. Sue and Krista Varantola (1998). Language learners using dictionaries: The final report on the euralex / aila research projekt on dictionary use. In B. T. Sue Atkins (ed.), *Using Dictonaries. Studies of Dictionary Use by Language Learners and Translators*, 21–81, Tübingen.

Engelberg, Stefan and Lothar Lemnitzer (2001). *Lexikographie und Wörterbuchbenutzung*. Tübingen: Stauffenburg.

Geeb, Franziskus (2001). Sprache und datenverarbeitung. In *Sprache und Datenverarbeitung*, volume 2, 27–61.

Harras, Gisela, Edeltraud Winkler, Sabine Erb, and Kristel Proost (2004). *Handbuch deutscher Kommunikationsverben. Teil 1: Wörterbuch (= Schriften des Instituts für Deutsche Sprache)*. Berlin / New York.: de Gruyter.

Haß, Ulrike (ed.) (2005). *Grundfragen der elektronischen Lexikographie. Elexiko – das Online-Informationssystem zum deutschen Wortschatz*, volume 12 of *Schriften des Instituts für Deutsche Sprache*. Berlin / New York.: deGruyter.

Herberg, Dieter, Michael Kinne, and Doris Steffens (2004). *Neuer Wortschatz. Neologismen der 90er Jahre im Deutschen. Unter Mitarbeit von Elke Tellenbach und Doris al-Wadi*. Berlin / New York: deGruyter.

Höhne, Steffen (1991). Die rolle des wörterbuchs in der sprachberatung: Eine sekundäranalyse zur wörterbuchbenutzungsforschung. *Zeitschrift für Germanistische Linguistik* 19:293–321.

Klosa, Annette (ed.) (2008). *Lexikografische Portale im Internet. (=OPAL-Sonderheft 1/2008)*. Mannheim: OPAL Sonderheft 1/2008, http://www.ids-mannheim.de/pub/laufend/opal/.

Klosa, Annette, Ulrich Schnörch, and Petra Storjohann (2006). Elexiko - a lexical and lexicological, corpus-based hypertext information system at the institut für deutsche sprache. In Carla Marello et al. (ed.), *Proceedings of the 12th EURALEX International Congress (Atti del XII Congresso Internazionale di Lessicografia), EURALEX 2006*, volume 1, 425–430, Turin, Italy.

Kunze, Claudia and Lothar Lemnitzer (2007). *Computerlexikographie. Eine Einführung*. Tübingen.: Narr.

Kämper, Heidrun (2005). *Der Schulddiskurs in der frühen Nachkriegszeit. Ein Beitrag zur Geschichte des sprachlichen Umbruchs nach 1945*, volume 78 of *Studia Linguistica Germanica*. Berlin / New York: deGruyter.

Lemnitzer, Lothar (2001). Das internet als medium für die wörterbuchbenutzungsforschung. In Ingrid Lemberg, Bernhard Schröder, and Angelika Storrer (eds.), *Chancen und Perspektiven computergestützter Lexikographie*, 247–254, Tübingen: Niemeyer.

Měchura, Michal Boleslav (2008). Giving them what they want: Search strategies for electronic dictionaries. In *Proceedings of the 13th EURALEX International Congress. Euralex 2008*, Barcelona, Spain, cD-ROM.

Müller, Wolfgang (1991). Einige problematische dudenbenutzungssituationen. ein florileg aus den sprachanfragen. In Gerhard Augst and Burkhard Schaeder (eds.), *Rechtschreibwörterbücher in der Diskussion*, 335–361, Frankfurt/M.

Müller-Spitzer, Carolin (2006). Das konzept der inhaltsstruktur. ein ausschnitt aus einer neuen konzeption für die modellierung lexikografischer daten. In *OPAL - Online publizierte Arbeiten zur Linguistik 2*, http://www.ids-mannheim.de/pub/laufend/opal/privat/opal06-2.html.

Müller-Spitzer, Carolin (2007a). Das elexiko-portal: Ein neuer zugang zu lexikografischen arbeiten am institut für deutsche sprache. In Georg Rehm, Andreas Witt, and Lothar Lemnitzer (eds.), *Datenstrukturen für linguistische Ressourcen und ihre Anwendungen*, 179–188.

Müller-Spitzer, Carolin (2007b). *Der lexikografische Prozess. Konzeption für die Modellierung der Datenbasis*. Tübingen: Narr.

Müller-Spitzer, Carolin (2007c). Vernetzungsstrukturen lexikografischer daten und ihre xml-basierte modellierung. In *Hermes*, volume 38, 137–171.

Ripfel, Martha (1989). Ergebnisse einer befragung zur benutzung ein- und zweisprachiger wörterbücher. In *Lexicographica*, volume 5, 178–201.

Scherer, Tanja (2008). *Umsetzung von Zugriffsstrukturen bei Online-Wörterbüchern*. Master's thesis, University Mannheim.

Schlaps, Christiane (2007). *Grundfragen der elektronischen Lexikographie. Elexiko – das Online-Informationssystem zum deutschen Wortschatz*. Berlin, New York: deGruyter, short review, in: Lexicographica 22, 72-94.

Schryver, Gilles-Maurice De (2003). Lexicographer's dreams in the electronic-dictionary age. *International Journal of Lexicography* 16(2):143–199.

Schryver, Gilles Maurice De and David Joffe (2004). On how electronic dictionaries are really used. In Geoffrey Williams and Sandra Vessier (eds.), *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004*, volume 1, 187–196, Lorient, France.

Schryver, Gilles Maurice De and Daan J. Prinsloo (2000). Dictionary-making process with the "simultaneous feedback" from the target users to the compilers. In Ulrich Heid, Stefan Evert, Egbert Lehmann, and Christian Rohrer (eds.), *Proceedings of the Ninth EURALEX International Congress. Euralex 2000*, volume 1, 197–209, Stuttgart, Germany.

Schumacher, Helmut, Jacqueline Kubczak, and Renate Schmidt (2004). *VALBU – Valenzwörterbuch deutscher Verben*. Tübingen: Narr.

Storrer, Angelika (2001). Digitale wörterbücher als hypertexte: Zur nutzung des hypertextkonzepts in der lexikographie. In Ingrid Lemberg, Bernhard Schröder, and Angelika Storrer (eds.), *Chancen und Perspektiven computergestützter Lexikographie*, 53–69, Tübingen: Niemeyer.

Thielen, Christine, Elisabeth Breidt, and Helmut Feldweg (1998). Compass. ein intelligentes wörterbuchsystem für das lesen fremdsprachiger texte. In Angelika Storrer and Bettina Harriehausen (eds.), *Hypermedia für Lexikon und Grammatik*, 173–194, Tübingen: Narr.

Wang, Weiwei (2001). *Zweisprachige Fachlexikographie. Benutzungsforschung, Typologie und mikrostrukturelle Konzeption*. Frankfurt/M.

Wiegand, Herbert Ernst (1998). *Wörterbuchforschung. Untersuchungen zur Wörterbuchbenutzung, zur Theorie, Geschichte, Kritik und Automatisierung der Lexikographie*, volume 1. Berlin / New York: deGruyter.

The Kicktionary revisited

Thomas Schmidt

Abstract. This paper presents a concept for the compilation of an electronic football dictionary through a community-driven process in a web-based lexicographic environment. It departs from an existing resource, the Kicktionary, in which English, French and German lexical units of football language were analysed according to the FrameNet and WordNet approaches and illustrated with annotated example sentences from a corpus of authentic football texts. Based on the experience gained in the construction of this resource, a suggestion is made for a web-based environment in which amateur and professional lexicographers can work together to produce a quantitatively and qualitatively improved and extended version of the Kicktionary.

1 Introduction

The Kicktionary (Schmidt 2006, 2007, 2008a,b) is a multilingual electronic dictionary of football language. It contains information about roughly 2,000 lexical units (henceforth: LUs) in three languages: English, French and German. Each LU is illustrated by a number of example sentences from a (partly parallel) corpus of football match reports. Using an approach inspired by the theory of frame semantics and the FrameNet project (Boas 2006; Fillmore 1977a,b, 1982; Fillmore et al. 2003; Petruck 1996; Ruppenhofer et al. 2006), the LUs are organised into a hierarchy of frames and scenes, and their example sentences are annotated for the corresponding entities (frame elements, supports). Furthermore, WordNet's concept of semantic relations (Fellbaum 1990, 1998) is used to establish a second, independent organisation of LUs into synsets and hierarchies of hyponyms, meronyms and troponyms. Translation equivalence between LUs in different languages is established by extending the notion of "synonymy" to the multilingual case.

The Kicktionary was developed in a one-year project, more or less by a single person. While it demonstrates several of the advantages that are typically associated with computer-assisted (or computer-based) lexicography – such as the independence of space restrictions, the possibility of making non-linear macro-structures navigable, and the flexible linking of dictionary and corpus data (see also Storrer 2001) – it has by no means tapped the full potential of these methods.

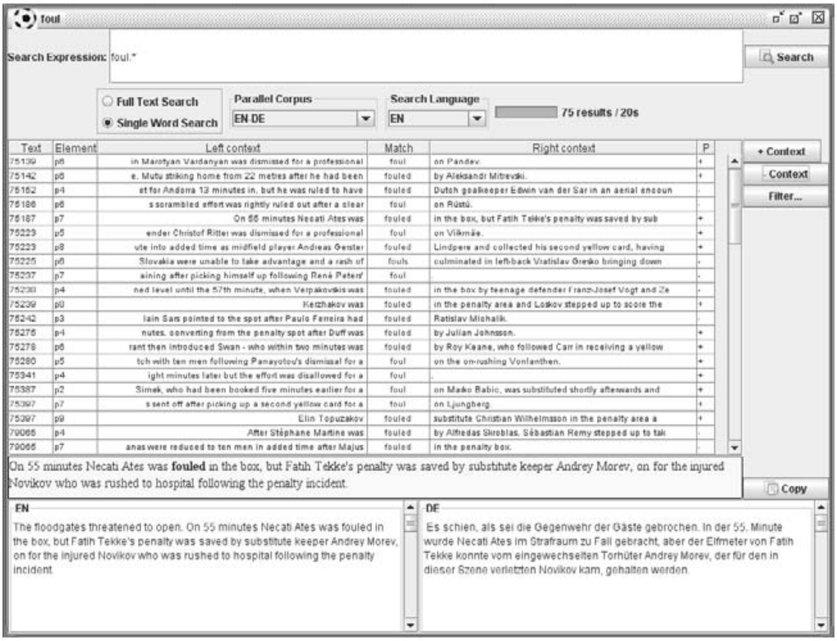
This paper aims to explore one further option of using computers for building and using lexicographic resources: the possibility to share the lexicographic work among an open community of amateur and professional lexicographers, who use a common technical infrastructure for their work, but are otherwise not, or only loosely, organised. In other words: this paper discusses a concept for the compilation of an electronic dictionary through a community-driven process in a web-based lexicographic environment. The Kicktionary in its present form is used as a starting point for this discussion. As a ready-to-use resource with a substantial amount of material in different languages, it can help to define and delimit different tasks as well as to recognize and assess potential difficulties in the resource construction. Section 2 discusses such general considerations before the concept for a web-based environment is formulated in more detail in section 3.

2 General considerations

2.1 Football vocabulary

Football vocabulary has proven, in several ways, to be a convenient area for testing new methods in lexicography. First, in contrast to general vocabulary, a football match with its limited set of actions and actors defines relatively clear boundaries to the set of lexical items to be investigated. As work on the Kicktionary has shown, a list of no more than 1,000 lexical units will thus cover a substantial part of the relevant vocabulary in a given language.[1] Second, in contrast, to most "real" specialist languages, football vocabulary exhibits a large degree of stylistic variation and other lexicographically interesting phenomena, including fine-grained semantic distinctions, an abundance of synonymous and near-synonymous items, many idiomatic expressions, as well as lexical gaps and other cross-lingual phenomena. Many of these interesting phenomena have already been described in great detail in the linguistic literature (e.g. Gross 2002; Seelbach 2001). Third, football texts are easily available in great number, because football as a popular sport receives a broad coverage in all types of media. Moreover, international competitions (like the Champions League or the World Cup) provide comparable reports of one and the same match in different languages and can thus become the starting point for harvesting parallel corpora. Fourth (and maybe most importantly for community-driven dictionary making), there is a large number of proficient "speakers" of football language who may be motivated to make some contribution to the resource.

[1] Most commercially available (printed) football dictionaries (e.g. Colombo et al. 2006 or Yildirim 2006) cover less than 400 lexical units. Of course, the number of relevant items is still potentially much higher than 1000, but I think it is safe to claim that a vocabulary of 1000 lexical units will contain words for all important concepts in and around a football match, and will also cover a fair amount of stylistic variation.



Search Expression: foul.*

○ Full Text Search
● Single Word Search

Parallel Corpus: EN DE

Search Language: EN

75 results / 20s

Text	Element	Left context	Match	Right context	P
75139	p8	in Marotyan Vardanyan was dismissed for a professional	foul	on Pandev.	+
75142	p6	e. Mutu striking home from 22 metres after he had been	fouled	by Aleksandr Mitrevski.	+
75162	p4	et for Andorra 13 minutes in, but he was ruled to have	fouled	Dutch goalkeeper Edwin van der Sar in an aerial encoun	-
75186	p6	s scrambled effort was rightly ruled out after a clear	foul	on Rüstü.	-
75187	p7	On 56 minutes Necati Ates was	fouled	in the box, but Fatih Tekke's penalty was saved by sub	+
75223	p5	ender Christof Ritter was dismissed for a professional	foul	on Vilkmäe.	+
75223	p8	ute into added time as midfield player Andreas Gerster	fouled	Lindpere and collected his second yellow card, having	+
75225	p6	Slovakia were unable to take advantage and a rash of	fouls	culminated in left-back Vratislav Gresko bringing down	-
75237	p7	aining after picking himself up following René Peters'	foul		-
75230	p4	ned level until the 57th minute, when Verpakovskis was	fouled	in the box by teenage defender Franz-Josef Vogt and Ze	-
75239	p0	Kerzhakov was	fouled	in the penalty area and Loskov stepped up to score the	+
75242	p3	Iain Sars pointed to the spot after Paulo Ferreira had	fouled	Ratislav Michalik.	-
75276	p4	nutes, converting from the penalty spot after Duff was	fouled	by Julian Johnsson.	+
75278	p6	rant then introduced Swan - who within two minutes was	fouled	by Roy Keane, who followed Carr in receiving a yellow	+
75280	p5	tch with ten men following Panayotou's dismissal for a	foul	on the on-rushing Vonlanthen.	+
75341	p4	ight minutes later but the effort was disallowed for a	foul	.	+
75387	p2	Simek, who had been booked five minutes earlier for a	foul	on Marko Babic, was substituted shortly afterwards and	+
75397	p7	s sent off after picking up a second yellow card for a	foul	on Ljungberg.	+
75397	p9	Elin Topuzakov	fouled	substitute Christian Wilhelmsson in the penalty area a	+
79066	p4	After Stéphane Martine was	fouled	by Alfredas Skroblas, Sébastian Remy stepped up to tak	-
79066	p7	anas were reduced to ten men in added time after Majus	fouled	in the penalty box.	-

On 55 minutes Necati Ates was **fouled** in the box, but Fatih Tekke's penalty was saved by substitute keeper Andrey Morev, on for the injured Novikov who was rushed to hospital following the penalty incident.

EN

The floodgates threatened to open. On 55 minutes Necati Ates was fouled in the box, but Fatih Tekke's penalty was saved by substitute keeper Andrey Morev, on for the injured Novikov who was rushed to hospital following the penalty incident.

DE

Es schien, als sei die Gegenwehr der Gäste gebrochen. In der 55. Minute wurde Necati Ates im Strafraum zu Fall gebracht, aber der Elfmeter von Fatih Tekke konnte vom eingewechselten Torhüter Andrey Morev, der für den in dieser Szene verletzten Novikov kam, gehalten werden.

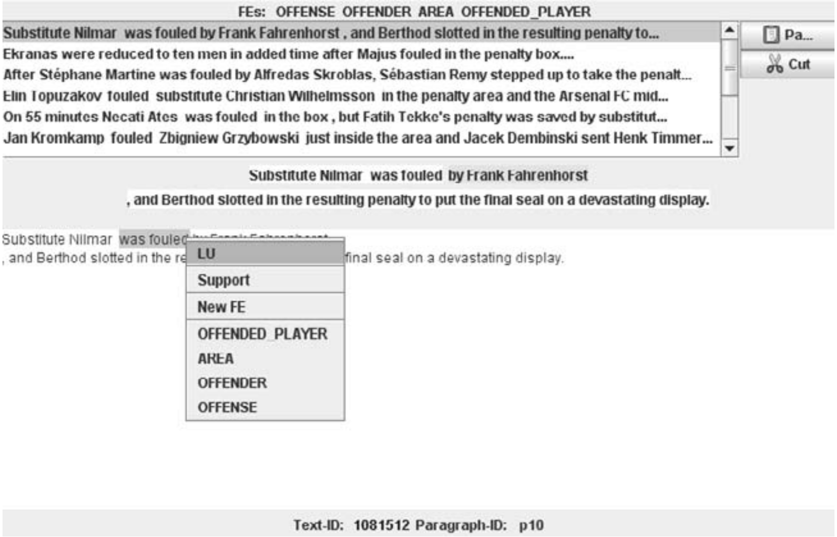
Figure 1. Query tool used for the Kicktionary

2.2 Dictionary construction workflow

The central instrument for building the Kicktionary was a combination of a query tool and an annotation tool. The query tool allows the user to query the resource for a regular expression (e.g. "[Pp]enalt(ylies)" to find all variants of the lexical unit "penalty"). Query results are initially presented in a keyword-in-context (KWIC) concordance. For each item in the KWIC concordance, additional context can be displayed by clicking on it. A double click will display aligned segments in parallel texts if they exist.

If an example sentence from this concordance is judged suitable for inclusion in the resource, it can be selected, assigned to a LU, and transferred for processing to the annotation tool. In order not to lose the link to the whole text, an exact pointer to the corpus location is stored with the sentence. The annotation tool then allows the user to mark certain sections of the example sentence and to assign annotation values (here: frame element names etc.) to them.

This simple process, which basically does no more than support the manual excerption and annotation of example sentences for a given LU, has proven very effective and efficient for gradually extending the list of LUs and their corpus examples,



FEs: OFFENSE OFFENDER AREA OFFENDED_PLAYER

Substitute Nilmar was fouled by Frank Fahrenhorst , and Berthod slotted in the resulting penalty to...
Ekranas were reduced to ten men in added time after Majus fouled in the penalty box....
After Stéphane Martine was fouled by Alfredas Skroblas, Sébastian Remy stepped up to take the penalt...
Elin Topuzakov fouled substitute Christian Wilhelmsson in the penalty area and the Arsenal FC mid...
On 55 minutes Necati Ates was fouled in the box , but Fatih Tekke's penalty was saved by substitut...
Jan Kromkamp fouled Zbigniew Grzybowski just inside the area and Jacek Dembinski sent Henk Timmer...

Substitute Nilmar was fouled by Frank Fahrenhorst
, and Berthod slotted in the resulting penalty to put the final seal on a devastating display.

Substitute Nilmar was fouled
, and Berthod slotted in the re final seal on a devastating display.

Text-ID: 1081512 Paragraph-ID: p10

Figure 2. Annotation tool used for the Kicktionary

i.e. for producing the basic building blocks of the lexical resource. The access which the query tool offers to the larger context and to aligned parallel text segments in other languages greatly facilitates the analysis of the lexical unit under investigation. Revisions of existing material, i.e. cases where later analyses made it necessary to modify example selections or annotations of earlier phases, were largely unproblematic on this level.

Likewise, establishing synonymy or translation equivalence relations between two given lexical units was straightforward in most cases. Of course, individual decisions in this area can become difficult (e.g. “are *to rifle (a ball)* and *to blast (a ball)* really synonymous or do their meanings differ by a nuance?” or “is *to thrash (an opponent)* in English really the same as *(einen Gegner) deklassieren* in German?”), but the difficulty of such decisions did not affect the overall workflow, because revising a single synonymy link has a local effect only.

However, the workflow was much more complex and less efficient when it came to higher level analyses, i.e. to the construction of the scenes-and-frames hierarchy and the establishment of semantic relations between sets of synonymous lexical units. On this level, difficulties of the following types were encountered (see Schmidt 2008a for a more detailed discussion):

- drawing boundaries between the scenes of a football match (e.g.: “is the kick-off after a goal still part of the ‘goal’ scene?”)

- assigning lexical units to one (and only one) existing scene (e.g.: “does the LU *free-kick* belong to the ‘shot’ scene or is it better characterised as being the last stage of a ‘foul’ scene?”)

- finding the right degree of abstraction for a frame’s core meaning (e.g.: “do the verbs *to block (a ball)* and *to parry (a ball)* belong to the same frame (because they both denote the action of intervening with a shot) or do they belong to different frames (because they are typically carried out by different actors, namely a defender and a goalkeeper, respectively)?”)

- applying frame-semantic principles to the description of nouns whose main function is to denote persons and objects (like *goalkeeper*, *substitute*, *byline*, *penalty area*) rather than to describe processes or activities (like *shot*,*free-kick*, *pass*)

As the resource grew in terms of LUs and example sentences, frequent and far-reaching adaptations of the existing higher-level structures thus became necessary, which, unlike modifications at lower levels, were difficult to oversee and which required a careful consideration and weighing-up of partly conflicting empirical evidence. In the end, the most helpful criteria for deciding such cases were of a pragmatic nature: many issues could be resolved by taking into account economy and balance of the data structure (e.g. “don’t make frames too small or too big”) and by considering which organization of the dictionary would be maximally transparent to a user.

In an environment where many users work together on a resource, these differences are crucial: while the compilation and stepwise extension of the basic building blocks (i.e. LUs with examples and synonymy and translation equivalence relations) of the dictionary is a task which may indeed be accomplished in a loosely organised community, building the higher level structure may require additional forms of coordination and supervision. Moreover, the more lexical units exist, the easier or more well-grounded decisions about higher level structures can be taken. The general approach for the resource described here should therefore be a bottom-up process in which the empirical investigation of lexical units determines the building of scenes and frames, rather than the other way around.[2]

[2] As Patrick Hanks notes in a discussion on the lexicography mailing list [http://groups.yahoo.com/group/lexicographylist/], this approach might also be preferable on methodological grounds to the one used by FrameNet: “FrameNet proceeds frame by frame, not word by word. This may seem a trivial point, but it isn’t. Although FrameNet uses empirical data, it does not use an empirical methodology.”

2.3 Quantitative and qualitative improvements for the Kicktionary

In its current state, the Kicktionary contains roughly 2,000 lexical units in the three languages, illustrated by a little more than 8,000 example sentences, extracted from a corpus of about 1,5 million tokens (1 million for German, 250,000 for both English and French). Obviously, there are several ways of quantitatively extending the resource:

- using larger corpora to find more lexical units (possibly with more stylistic variation),
- using larger corpora to find more example sentences (possibly with more syntactic variation),
- using corpora from other languages to make the dictionary more multilingual.

In terms of qualitative improvements, the following three issues seem to be most important:

- the provision of definitions for individual lexical units,
- a more explicit method for building the scenes and frames hierarchy, to be evaluated with respect to its utility and usability for the dictionary user,
- a more careful control of cross-lingual links (translation equivalences) involving native speakers of both languages.

Especially for the last points of the respective lists, a web-based environment in which contributors from different places can access and work on the resource asynchronously is a very helpful, maybe even indispensable, instrument.

2.4 Potential users and contributors

The concept presented in this paper is based on the assumption that there exists a community willing to use and contribute to an electronic multilingual football dictionary. Because of this, and since the choice of an application scenario can have far-reaching consequences for the design of such a resource, it may be useful to attempt to draw a clearer picture of potential user and contributor groups. The goal of the Kicktionary was and remains to produce a lexical resource usable by humans for purposes of understanding, translating or otherwise paraphrasing texts in the domain of football. In contrast to much work carried out by FrameNet and by related projects, the Kicktionary does thus not claim to make contributions to fields like machine translation, question answering or other subareas of natural language processing or artificial intelligence. That said, the following potential user groups come to mind:

- The monolingual layman who wants to learn and understand the football vocabulary of his own language. He or she can be expected to have a high proficiency in the respective language, but little knowledge of football. A clear and plausible organisation of the vocabulary as well as the possibilities frame semantics offers for linking linguistic with encyclopaedic knowledge should be of special importance to this user group.

- The bilingual fan who wants to understand texts about football in another language and who wants to have basic conversations about football in a foreign language. He or she can be expected to have a low to medium proficiency in the target language, and a good knowledge of football. His demands on the level of linguistic detail of the dictionary will be moderate – he will typically want quick access to a translation or an example sentence for a certain word, but will be less interested in systematically exploring the whole variation of the vocabulary.

- The bilingual professional, i.e. a player, a coach, a journalist or an official who needs to talk or write regularly and professionally about football in a language different from his mother tongue. He or she can be expected to have medium to good knowledge of the target language and an expert's knowledge of football. His demands on the level of linguistic detail of the dictionary will be higher than those of the bilingual fan, i.e. he will be interested in fine semantic and stylistic distinctions within and between languages, he may need instruments supporting him in the construction of professional texts and translations etc.

As for potential contributors, it should be sufficient to distinguish the following two groups:

- Football 'experts' (i.e. the fans or professionals from the above user groups) with an interest in lexicography

- Lexicographers with an interest in football

Although there is certainly an intersection between the two groups, i.e. lexicographers with expert knowledge of football, a contributor will normally be either the one or the other. Moreover, given the number of languages involved, it is very unlikely that any single person is enough of an expert in all areas and languages to act as a supervisor for every detail of the dictionary. A special challenge and appeal in the collaborative construction of a resource is therefore to find optimal ways of coordinating and exploiting the exchange of knowledge between different contributor groups.

3 A web-based environment for a collaborative construction of the Kicktionary

Based on the considerations of the preceding section, this section presents a concept for a web-based environment in which different contributor groups can work together to produce an improved and extended electronic multilingual football dictionary.

3.1 Design principles

The existing Kicktionary will be used as a point of reference, though not necessarily as an exact blueprint, for its improved and extended version. That is, the web-based environment is aimed at constructing an electronic resource in which individual lexical units are illustrated by example sentences from an authentic corpus, and in which FrameNet and WordNet-like structures are used for a higher-level organisation of the lexical data. However, as has been argued in section 2.2, it is important to clearly separate the construction of the basic building blocks of the dictionary from the higher-level organisation in such a way that the latter does not predetermine the former. In other words: to ensure that different users can make individual contributions to the resource without always having to take into account its overall structure, a bottom-up workflow must be possible which proceeds LU by LU (rather than, say, frame by frame or scene by scene). In that sense, contributions on different levels with increasing complexity should be possible, depending on how much time and competence an individual contributor can offer:

- On level 0, a contributor can make a comment on an existing entry or a suggestion for an additional entry which he finds missing in the resource.

- On level 1, a contributor can decide to integrate a new lexical unit into the resource by writing a definition for it, selecting one or several suitable example sentences from the corpus and marking the LU in them.

- On level 2, a contributor can establish a synonym link between two LUs of the same language or a translation equivalence link between two LUs in different languages.

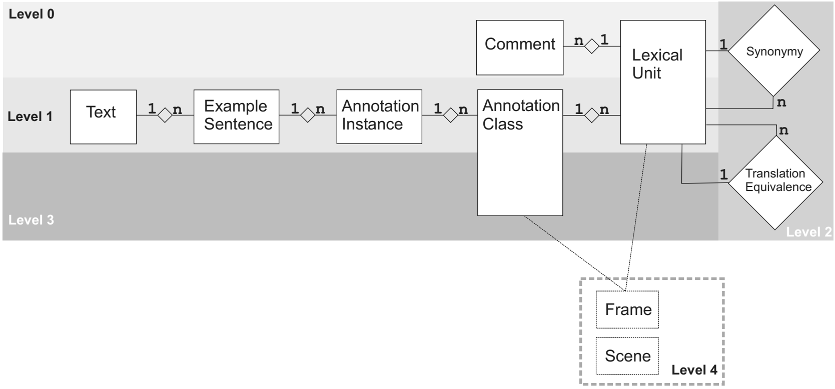


Figure 3. Database architecture

- On level 3, a contributor can further annotate example sentences of a given LU by marking their arguments and assigning them to semantic roles. These annotations will then constitute the base material for defining and describing frames and scenes.

- On level 4, the higher level structures, i.e. hierarchies of scenes and frames and hierarchies of semantically related LUs are built.

Most contributors will probably be content to make contributions on levels 0 to 2. Levels 3 and 4 require advanced lexicographic knowledge, and should therefore be restricted to contributors who have some qualification in this area, and who are ready to get involved in a discussion on the best principles for identifying and defining higher level entities. However, feedback in the form of comments should be possible for every user on all levels – this can also be seen as a first form of evaluation of the resulting resource.

3.2 Implementation

The actual implementation will be done using standard web technology. Corpus data and lexicographic data will be kept in a relational database. Figure 3 sketches the basic components of the database and their relationships.

Processing steps are differentiated according to levels and affect different parts of the architecture:

- Candidates for lexical units are suggested at level 0 and confirmed at level 1.

EXAMPLE SENTENCE

ID	Text	i_1	i_2	Text
S#1	TEXT#1	154	255	*Acht Minuten vor Schluß der ersten Hälfte setzte Bogdani einen Kopfball am türkischen Gehäuse vorbei.*

ANNOTATION CLASS

ID	Type	Value	LU
AC#1	Lexical_Unit	Kopfball	LU#1
AC#3	Argument	SHOOTER	LU#1
AC#4	Argument	TARGET	LU#1

ANNOTATION INSTANCE

ID	Sentence	i_1	i_2	Text	Class
AI#1	S#1	65	72	*Kopfball*	AC#1
AI#2	S#1	50	57	*Bogdani*	AC#3
AI#3	S#1	73	101	*am türkischen Gehäuse vorbei*	AC#4

LEXICAL UNIT

ID	Lemma
LU#1	Kopfball
LU#2	header
LU#3	Kopfstoß

SYNONYMY

LU1	LU2
LU#1	LU#3

TRANSLATION EQUIV

LU1	LU2
LU#1	LU#2

Figure 4. Annotating an example for the LU *Kopfball* and assigning a synonym and a translation equivalent

- Each lexical unit defines a set of annotation classes. On level 1, the only annotation class is a marker for the lexical unit itself, i.e. its inflected form inside a corpus sentence. This may be a discontinuous unit as, for example, in German particle verbs (e.g. “er spielte den Ball ab” for the LU *abspielen*). On level 3, further annotation classes for marking and denominating argument classes (e.g. SHOOTER and TARGET for the LU *Kopfball*) can be defined and applied.

- The actual annotation is done through an annotation instance, which is realised as a standoff pointer from an annotation class into the corpus text. In that way, problems with respect to overlapping annotations are avoided, and it becomes possible to annotate one and the same sentence for several lexical units.

- Synonymy and translation equivalence (on level 2) are treated as pairwise assignments of lexical units. On the application level, transitive closure can be used to infer equivalence relations that are not explicitly represented on the data level. Thus, for instance, from the fact that *Kopfball* is a synonym of *Kopfstoß* and that header is a translation of *Kopfball*, it can be inferred that *Kopfstoß* also translates as *header*.

- Frames and scenes (on level 4) are treated as generalisations over a set of lexical units and their annotation classes. For instance, constructing a SHOOT frame or scene will involve a decision to declare the individual argument markers SHOOTER of LUs like *Kopfball*, *Schuß*, *schießen*, *abziehen* etc. as instances of one and the same frame element marker.

Figure 4 exemplifies this for levels 0 to 3.

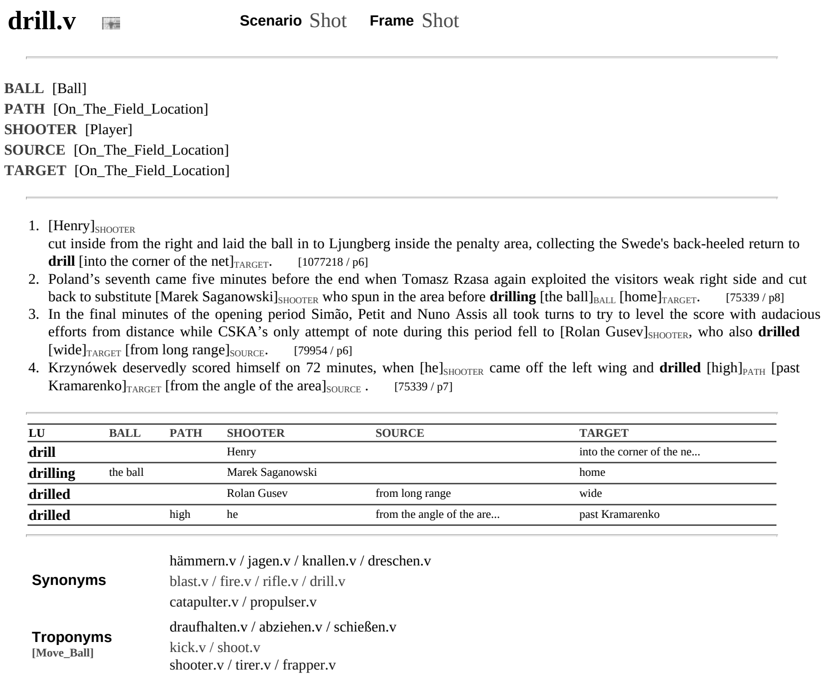
Corpus data are currently available for the major European languages German, English, French, Spanish, Italian and Portuguese as well as for Russian, Japanese and Korean. There are at least 500,000 tokens of written text for each language. All texts are either after-match reports or minute-by-minute reports as used in news tickers and RSS feeds. Parts of the corpus are parallel, i.e. texts are translations of one another. A rough alignment of these texts on the paragraph level[3] or on the level of individual ticker entries will be done and made available for the lexicographic analysis. The architecture should, in principle, be open to integrate additional texts from these or other languages. The web interface, generated from the database with the help of appropriate technologies (PHP, Ajax and/or Java), will present individual LU entries in an integrated display giving all existing information (annotated example sentences, synonyms, translation equivalents etc.) about them. The Kicktionary's present (read-only) user interface (see figure 5) can serve as a point of orientation for the design of the interactive user interface.

Starting from this kind of display or from a simple list of LUs, contributors must be enabled to modify or supplement the resource. Their contributor level (see above) will determine what kinds of modification they are allowed. For levels 1 and 3, a query and annotation interface similar to the ones used in the construction of the original Kicktionary (see above) will be necessary. For level 2, an interface allowing the establishment of links between two existing LUs in the resource must be provided. This can be done, for instance, in the form of two parallel lists. Level 0 can be realised by appropriate comment fields for every type of existing information in the resource. What type(s) of interface(s) are needed for level 4 remains to be investigated.

4 Conclusion

In this paper, I have suggested a concept for the compilation of an electronic dictionary through a community-driven process in a web-based environment, and I have sketched an architecture for the actual implementation of such an environment. The architecture reflects an essential observation gained from the construction of the original Kicktionary, namely that FrameNet- and WordNet-like resources contain information of highly different complexity. The basic building blocks, i.e. lexical units, their definitions and annotated example sentences, lend themselves comparatively easily to a distributed, loosely organised construction process. The same holds, but maybe already to a lesser degree, for the establishment of synonymy and translation equivalence relations between lexical units. Higher level structures, i.e. frames, scenes and concept hierarchies, on the other hand, may require a more thoroughly

[3] Texts usually consist of no more than 10 paragraphs. Aligning parapgraphs according to their position in the text will yield a correct alignment in more than 80% of all cases which has proven to be sufficiently precise for the cross-linguistic lexicographic analysis



drill.v **Scenario** Shot **Frame** Shot

BALL [Ball]
PATH [On_The_Field_Location]
SHOOTER [Player]
SOURCE [On_The_Field_Location]
TARGET [On_The_Field_Location]

1. [Henry]$_{\text{SHOOTER}}$ cut inside from the right and laid the ball in to Ljungberg inside the penalty area, collecting the Swede's back-heeled return to **drill** [into the corner of the net]$_{\text{TARGET}}$. [1077218 / p6]
2. Poland's seventh came five minutes before the end when Tomasz Rzasa again exploited the visitors weak right side and cut back to substitute [Marek Saganowski]$_{\text{SHOOTER}}$ who spun in the area before **drilling** [the ball]$_{\text{BALL}}$ [home]$_{\text{TARGET}}$. [75339 / p8]
3. In the final minutes of the opening period Simão, Petit and Nuno Assis all took turns to try to level the score with audacious efforts from distance while CSKA's only attempt of note during this period fell to [Rolan Gusev]$_{\text{SHOOTER}}$, who also **drilled** [wide]$_{\text{TARGET}}$ [from long range]$_{\text{SOURCE}}$. [79954 / p6]
4. Krzynówek deservedly scored himself on 72 minutes, when [he]$_{\text{SHOOTER}}$ came off the left wing and **drilled** [high]$_{\text{PATH}}$ [past Kramarenko]$_{\text{TARGET}}$ [from the angle of the area]$_{\text{SOURCE}}$. [75339 / p7]

LU	BALL	PATH	SHOOTER	SOURCE	TARGET
drill			Henry		into the corner of the ne...
drilling	the ball		Marek Saganowski		home
drilled			Rolan Gusev	from long range	wide
drilled		high	he	from the angle of the are...	past Kramarenko

Synonyms
hämmern.v / jagen.v / knallen.v / dreschen.v
blast.v / fire.v / rifle.v / drill.v
catapulter.v / propulser.v

Troponyms
[Move_Ball]
draufhalten.v / abziehen.v / schießen.v
kick.v / shoot.v
shooter.v / tirer.v / frapper.v

Figure 5. LU display in the Kicktionary

planned and supervised approach. Consequently, the architecture proposed here is based on a bottom-up methodology in which many contributors carry out the bulk of the empirical corpus work before more complex structure is added to the resource. Hopefully, this manner of proceeding will not only lead to an interesting multilingual resource, but will also allow to draw some more general conclusions about new ways of carrying out lexicographic analyses.

References

Boas, Hans C. (2006). Semantic Frames as Interlingual Representations for Multilingual Lexical Databases. *International Journal of Lexicography* 4(18):445–478.

Colombo, Roberta, Klaus Heimeroth, Olivier Humbert, Michael Jackson, Frank Kohl, and Josep Ràfols (2006). *PONS Fußballwörterbuch*. Stuttgart: Ernst Klett Verlag.

Fellbaum, Christiane (1990). English Verbs as a Semantic Net. *International Journal of Lexicography* 3(4):278–301.

Fellbaum, Christiane (ed.) (1998). *WordNet - An Electronic Lexical Database*. MIT Press.

Fillmore, Charles J. (1977a). Scenes-and-Frames Semantics, Linguistic Structure Processing. In Antonio Zampolli (ed.), *Fundamental Studies in Computer Science*, 55–88, North Holland Publishing.

Fillmore, Charles J. (1977b). The Case for Case Reopened. In Peter Cole and Jerrold Saddock (eds.), *Syntax and Semantics 8: Grammatical Relations*, 59–82, New York: Academic Press.

Fillmore, Charles J. (1982). Frame Semantics. In *Linguistics in the Morning Calm*, 111–137, Seoul: Hanshin Publishing Company.

Fillmore, Charles J., Christopher Johnson, and Miriam R.L. Petruck (2003). Background to Framenet. *International Journal of Lexicography* 3(16):235–250.

Gross, Gaston (2002). Comment décrire une Langue de Spécialité? *Cahiers de Lexicologie: Revue Internationale de Lexicologie et Lexicographie* (80):179–200.

Petruck, Miriam R.L. (1996). Frame Semantics. In Jef Verschueren (ed.), *Handbook of Pragmatics*, Philadelphia: John Benjamnins.

Ruppenhofer, Josef, Michael J. Ellsworth, Miriam R.L. Petruck, and Christopher Johnson (2006). *FrameNet: Theory and Practice*. URL `http://framenet.icsi.berkeley.edu/book/book.html`.

Schmidt, Thomas (2006). Interfacing Lexical and Ontological Information in a Multilingual Soccer FrameNet. In *Proceedings of Ontolex*, Paris: ELRA.

Schmidt, Thomas (2007). The Kicktionary: A Multilingual Lexical Resource of the Language of Football. In Georg Rehm, Andreas Witt, and Lothar Lemnitzer (eds.), *Data Structures for Linguistic Resources and Applications*, 189–196, Tuebingen: Gunter Narr.

Schmidt, Thomas (2008a). The Kicktionary - A Multilingual Lexical Resource of Football Language. In Hans C. Boas (ed.), *Multilingual Framenets*, New York: De Gruyter.

Schmidt, Thomas (2008b). The Kicktionary: Combining Corpus Linguistics and Lexical Semantics for a Multilingual Football Dictionary. In Eva Lavric, Gerhard Pisek, Andrew Skinner, and Wolfgang Stadler (eds.), *The Linguistics of Football*, 11–23, Tuebingen: Gunter Narr.

Seelbach, Dieter (2001). Das Kleine Multilinguale Fußballlexikon. In Walter Bisang and Gabriela Schmidt (eds.), *Philologica et Linguistica. Historia, Pluralitas, Universitas. Festschrift für Helmut Humbach zum 80. Geburtstag am 4. Dezember 2001*, 323–350, Trier: Wissenschaftlicher Verlag.

Storrer, Angelika (2001). Digitale Wörterbücher als Hypertexte: Zur Nutzung des Hypertextkonzepts in der Lexikographie. In Ingrid Lemberg, Bernhard Schröder, and Angelika Storrer (eds.), *Chancen und Perspektiven Computergestützter Lexikographie. Hypertext, Internet und SGML/XML für die Produktion und Publikation digitaler Wörterbücher*, 88–104, Tübingen: Niemeyer.

Yildirim, Kaya (2006). *Fußballwörterbuch in 7 Sprachen*. Kauderwelsch, Bielefeld: Reise-Know-How Verlag Peter Rump GmbH.

List of Authors

Sabine Bartsch

Institut für Sprach-
und Literaturwissenschaft
Technische Universität Darmstadt
Hochschulstr. 1
D-64289 Darmstadt, Germany
`bartsch@linglit.tu-darmstadt.de`

Mark Buckley

Fachrichtung 4.7
Allgemeine Linguistik
Universität des Saarlandes
P.O. Box 15 11 50
D-66041 Saarbrücken, Germany
`buckley@coli.uni-sb.de`

Peter E. Clark

Boeing Research and Technology
PO Box 3707, m/s 7L-66
Seattle, WA 98124-2207, USA
`peter.e.clark@boeing.com`

Simon Clematide

Institut für Computerlinguistik
Universität Zürich
Binzmühlestr. 14
CH-8050 Zurich, Switzerland
`siclemat@cl.uzh.ch`

Irene Cramer

Institut für deutsche Sprache
und Literatur
Technische Universität Dortmund
Martin-Schmeißer-Weg 13
D-44227 Dortmund, Germany
`irene.cramer@uni-dortmund.de`

Jörg Didakowski

Digitales Wörterbuch
Berlin-Brandenburgische
Akademie der Wissenschaften
Jägerstr 21/22
D-10117 Berlin, Deutschland
`didakowski@bbaw.de`

Stefanie Dipper

Sprachwissenschaftliches Institut
Ruhr-Universität Bochum
Universitätsstr. 150
D-44780 Bochum, Germany
`dipper@linguistics.rub.de`

Kurt Eberle

Lingenio GmbH

Karlsruher Str. 10
D-69126 Heidelberg, Germany
`k.eberle@lingenio.de`

Kerstin Eckart

Institut für maschinelle
Sprachverarbeitung
Universität Stuttgart

Azenbergstr. 12
D-70174 Stuttgart, Germany
`eckartkn@ims.uni-stuttgart.de`

Richard Eckart

Institut für Sprach-
und Literaturwissenschaft
Technische Universität Darmstadt

Hochschulstr. 1
D-64289 Darmstadt, Germany
`eckart@linglit.tu-darmstadt.de`

Ana Fernández Montraveta

Departament de Filologia
Anglesa i de Germanística
Universitat Autònoma de Barcelona

Campus de la UAB
08193 Bellaterra, Spain
`ana.fernandez@uab.es`

Christiane Fellbaum

Cognitive Science Laboratory
Princeton University

Princeton, NJ 08540, USA
`fellbaum@princeton.edu`

Marc Finthammer

Fakultät für Mathematik
und Informatik
FernUniversität in Hagen

D-58084 Hagen, Germany
`marc@finthammer.de`

Fabienne Fritzinger

Institut für maschinelle
Sprachverarbeitung
Universität Stuttgart

Azenbergstr. 12
D-70174 Stuttgart, Germany
`fritzife@ims.uni-stuttgart.de`

Alexander Geyken

Digitales Wörterbuch
Berlin-Brandenburgische
Akademie der Wissenschaften
Jägerstr 21/22
D-10117 Berlin, Deutschland
`geyken@bbaw.de`

Susanne Hauptmann

Verlag C. H. Beck
Wilhelmstr. 9
D-80801 München, Germany
`susanne.hauptmann@beck.de`

Ulrich Heid

Institut für maschinelle
Sprachverarbeitung
Universität Stuttgart
Azenbergstr. 12
D-70174 Stuttgart, Germany
`heid@ims.uni-stuttgart.de`

Axel Herold

Digitales Wörterbuch
Berlin-Brandenburgische
Akademie der Wissenschaften
Jägerstr 21/22
D-10117 Berlin, Deutschland
`herold@bbaw.de`

Erhard Hinrichs

Abteilung Computerlinguistik
Seminar für Sprachwissenschaft
Universität Tübingen
Wilhelmstr. 19
D-72074 Tübingen, Germany
`eh@sfs.uni-tuebingen.de`

Jerry Hobbs

Information Science Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292, USA
`hobbs@isi.edu`

Marcin Junczys-Dowmunt

Instytut Językoznawstwa
Uniwersytet im. Adama Mickiewicza
Al. Niepodległości 4
61-874 Poznań, Poland
`junczys@amu.edu.pl`

Bryan Jurish

Deutsches Textarchiv
Berlin-Brandenburgische
Akademie der Wissenschaften

Jägerstr 21/22
D-10117 Berlin, Germany
jurish@bbaw.de

Birgit Kellner

Institut für Germanistik I
Universität Hamburg

Von-Melle-Park 6
D-20146 Hamburg, Germany
birgit.kellner@uni-hamburg.de

Manuel Kountz

Institut für maschinelle
Sprachverarbeitung
Universität Stuttgart

Azenbergstr. 12
D-70174 Stuttgart, Germany
kountzml@ims.uni-stuttgart.de

Timm Lehmberg

Institut für Germanistik I
Universität Hamburg

Von-Melle-Park 6
D-20146 Hamburg, Germany
timm.lehmberg@uni-hamburg.de

Joakim Lundborg

Institutionen för lingvistik
Stockholms universitet

S-106 91, Stockholm, Sweden
joakim.lungborg@gmail.com

Torsten Marek

Fachrichtung 4.7
Allgemeine Linguistik
Universität des Saarlandes

P.O. Box 15 11 50
D-66041 Saarbrücken, Germany
torsten.marek@gmx.net

Carolin Müller-Spitzer

Abteilung Lexik
Institut für Deutsche Sprache

P.O. Box 10 16 21
D-68016 Mannheim, Germany
mueller-spitzer@ids-mannheim.de

Reinhard Rapp

GRLMC
Facultat de Lletres
Universitat Rovira i Virgili

Pl. Imperial Tárraco 1
43005 Tarragona, Spain
reinhard.rapp@urv.cat

Thomas Schmidt

SFB 538 Mehrsprachigkeit
Universität Hamburg

Max-Brauer-Allee 60
D-22765 Hamburg, Germany
thomas.schmidt@uni-hamburg.de

Bettina Schrader

text & form GmbH

Oudenarder Str. 16
D-13347 Berlin, Germany
bettina_schrader@textform.com

Ingrid Schröder

Institut für Germanistik I
Universität Hamburg

Von-Melle-Park 6
D-20146 Hamburg, Germany
ingrid.schroeder@uni-hamburg.de

Lara Schwarz

Institut für Sprach-
und Literaturwissenschaft
Technische Universität Darmstadt

Hochschulstr. 1
D-64289 Darmstadt, Germany
schwarz.lara@googlemail.com

Elke Teich

Institut für Sprach-
und Literaturwissenschaft
Technische Universität Darmstadt

Hochschulstr. 1
D-64289 Darmstadt, Germany
teich@linglit.tu-darmstadt.de

Glòria Vázquez

Departament d'Anglés i Lingüística
Universitat de Lleida

Pl. Víctor Siurana 1
25003 Lleida, Spain
gvazquez@dal.udl.cat

Martin Volk

Institut für Computerlinguistik
Universität Zürich

Binzmühlestr. 14
CH-8050 Zurich, Switzerland
volk@cl.uzh.ch

Julia Weidenkaff

Institut für maschinelle
Sprachverarbeitung
Universität Stuttgart
Azenbergstr. 12
D-70174 Stuttgart, Germany
`weidenja@ims.uni-stuttgart.de`

Marion Weller

Institut für maschinelle
Sprachverarbeitung
Universität Stuttgart
Azenbergstr. 12
D-70174 Stuttgart, Germany
`wellermn@ims.uni-stuttgart.de`

Kai Wörner

SFB 538 Mehrsprachigkeit
Universität Hamburg
Max-Brauer-Allee 60
D-22765 Hamburg, Germany
`kai.woerner@uni-hamburg.de`

Magdalena Wolska

Fachrichtung 4.7
Allgemeine Linguistik
Universität des Saarlandes
P.O. Box 15 11 50
D-66041 Saarbrücken, Germany
`magda@coli.uni-sb.de`

Thomas Zastrow

Abteilung Computerlinguistik
Seminar für Sprachwissenschaft
Universität Tübingen
Wilhelmstr. 19
D-72074 Tübingen, Germany
`post@thomas-zastrow.de`

Index

alignment, 35, 38–48, 81, 165
annotation, 203
automatic corpus annotation, 69

bilingual alignment, 90
bilingual corpora, 147
bilingual dictionary, 84, 119, 151
Bulgarian dialectology, 112

chunking, 59, 124, 149
collaborative planning, 104
collocation, 119–131
collocation extraction, 131
CoMa metadata schema, 54
community-driven process, 220
compound noun splitting, 81–93
computational dialectometry, 108, 115
constraint grammar, 59–71
corpus annotation, 161

DAMSL, 99–100
data visualization, 107–116
dependency parser, 123, 124, 133
dependency structure, 17, 59–68, 126
Deutsches Wörterbuch (DWB), 27, 28
dialectology, 49, 108–116
dialectometric distance, 110
dialectometry, 110
dialects, 37–58
dialogue, 95–106
dialogue acts, 95
dialogue moves, 95–106
dictionary use, 205, 209
digital edition, 28
disambiguation, 133–145, 150
distance matrice, 108
distinguished discourse referent (DRF), 155
DWDS, 171, 193

EAGLES specification, 172
edit distance, 34
electronic archive, 113
electronic dictionary, 119, 210
electronic WDG (eWDG), 193
elexiko, 209, 217
Emdros, 12
EXMARaLDA, 54

feedback, 208
Festival, 28–30
finite state, 29, 35, 59
football vocabulary, 220
frame semantic, 219, 225
FrameNet, 223
FSA, 91
FST, 29–35
FUDRT, 144, 154

German dialects, 38, 48
GermaNet, 182–191
GermaNet Explorer, 183
GermaNet Pathfinder, 186
GIZA++, 81, 90, 149
glosses, 75, 182

High German, 49
historical text, 27

inflection classes, 173–178
information extraction (IE), 15
intelligent tutoring systems, 95
interview, 209, 215
IPA, 109
isoglosse, 107

JRC-Acquis parallel corpus, 82
juridical concept, 119

Kicktionary, 219–229
knowledge base, 73

LeActiveMath corpus, 98, 99
lemmatization, 171, 178
letter-to-sound (LTS), 28–32
lexical data, 109, 161
lexical semantic relations, 181
lexical unit (LU), 219–229
lexicographic data, 211, 219
lexicographic environment, 220
lexicography, 205–219
lexicon, 83, 170–179
lexicon expansion, 152

local dependency structures, 59
logical form (LF), 75
Low German, 49

machine translation (MT), 147
machine translation dictionaries, 147
medieval German text, 37–48
Middle German, 37
monolingual corpora, 151
morphological tools, 169
morphology, 28, 169–178
morphosemantic links, 77
multiword expression, 119

n-grams, 38–48, 147
NEGRA corpus, 3
nominalization, 133, 173
Northern German dialects, 52

OLIF, 169–178
online-dictionary, 214
orthographic form, 27
OWID, 205–216

parallel corpus, 39, 81, 84, 161
parallel treebanks, 3–13
parser, 15–25, 71, 75, 133
PASCAL RTE3 challenge, 74
PCFG, 17
Penn Treebank, 3
phonetic edit distance, 28
phonetic form, 28–34
Poleng lexicon, 88
probabilistic model, 86

recognizing textual entailment (RTE), 74
regional language, 49
rheme, 16

SAPIR, 75
semantic relations, 185
semiring, 60, 66
SFST, 171
similarity, 33–34, 37, 108, 114
SiN metadata schema, 56
SMOR, 123, 171
sortal paradox, 142
Spanish WordNet, 161
spoken language, 27
Stockholm TreeAligner, 12
STTS, 61, 123, 172
Syntactic Constraint Parser (SynCoP), 59–69
syntactic tagging, 59–71

TAGH morphology, 35, 170
task-oriented dialogue, 96
TAXI, 28–34
taxonomy, 78
taxonomy of dialogue moves, 106
TEI P5, 198
text understanding, 76
thematic progression, 16
theme identification, 15, 25
TIGER, 3–14
TIGER query language, 3–13
TRAINS corpus, 98, 102
translation relations, 158
treebank, 3–13
tutorial dialogues, 95–106

underspecified representations, 135–139, 144, 154
universal quantification, 3–14
unsupervised methods, 81, 91
Upper German, 37
user-adapted views, 205

VDM software, 107
visualization, 110
vocabulary extraction, 81

web, 81, 85
web service, 109, 179
web-based environment, 220
WFST, 60, 69
Wizard-of-Oz experiment, 97
word alignment, 92
word sense view, 194
WordNet, 73–80, 161–168, 181, 219

X-Sampa, 109, 110
XFST, 174